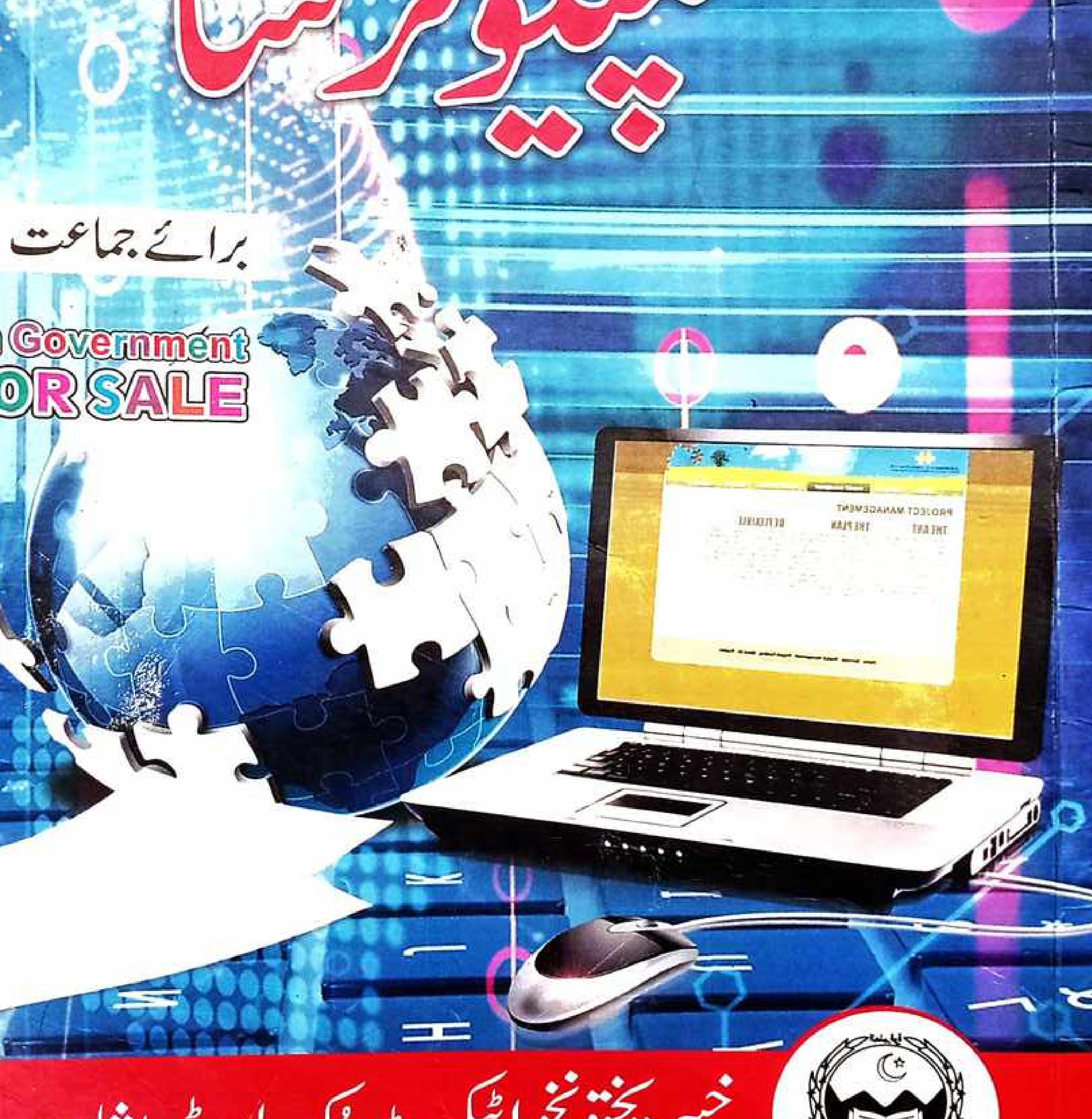
ٹیسٹ ایڈیشن

ٹیکسٹ بُک آف

کمپیوٹر سائنس

برائے جماعت دہم

Free From Government
NOT FOR SALE



خیبر پختونخوا ٹیکسٹ بُک بورڈ، پشاور

مومن تو آپس میں بھائی بھائی ہیں۔ پس اپنے دو بھائیوں میں صلح
کرا دیا کرو اور اللہ سے ڈرتے رہو تاکہ تم پر رحمت کی جائے۔
(سورۃ الحجرات: ۱۰)

Khyber Pakhtunkhwa Textbook Board
Phase – V, Hayatabad Peshawar
Phone: 091-9217714-15 Fax: 091-9217163, Website: www.kptbb.gov.pk

TEXTBOOK FEEDBACK FORM

S#	Statement	Remarks
1.	Overall the book is interesting and user friendly.	
2.	The language and content of the book is age / grade appropriate and the content is free of grammatical and punctuation errors.	
3.	Content is supported with examples from real life / culture.	
4.	Contents / texts are authentic and updated.	
5.	Pictures / diagrams / graphs / illustrations are informative, relevant and clear if not, then identify them.	
6.	Activities, projects and additional work is suggested for reinforcement of concepts.	
7.	Assessment achievements are thought provoking and comprise cognitive, psychomotor and effective skills.	
8.	The textbook is easy to be covered within academic year.	

Page No.	Observation/comments	Suggested amendment along with rationale

General Remarks: ____

____

Name & Address: ____

Contact No. & Email: ____

ٹیکسٹ بُک آف

کمپیوٹر سائنس

برائے جماعت دہم

X

خیبر پختونخوا ٹیکسٹ بک بورڈ، پشاور

جملہ حقوق بحق لیڈنگ بکس پبلشرز، پشاور محفوظ ہیں۔

منظور کردہ: نظامت نصاب و تعلیم اساتذہ خیبر پختونخوا، ایبٹ آباد
قومی نصاب بمطابق: 2009
مراسلہ نمبر: No.4449-51/S-04/TBB/Comp-X-U مورخہ: 30.07.2021

مصنف: محمد خالد، خالد اسد

مترجم: سعدیہ بصر

نگران نظر ثانی: گوہر علی خان، ڈائریکٹر، نصاب و تعلیم اساتذہ خیبر پختونخوا، ایبٹ آباد

فوکل پرسن: ذوالفقار خان، ایڈیشنل ڈائریکٹر، نظامت نصاب و تعلیم اساتذہ خیبر پختونخوا، ایبٹ آباد

نظر ثانی:

صلاح الدین ایوبی، اسسٹنٹ پروفیسر، گورنمنٹ پوسٹ گریجویٹ کالج نمبر ۱، ایبٹ آباد
فہیم احمد، لیکچرار، گورنمنٹ پوسٹ گریجویٹ کالج منڈیاں، ایبٹ آباد
سہیل انجم، ایس ایس ٹی (آئی ٹی)، گورنمنٹ ہائی سکول نمبر ۴، ایبٹ آباد
سفیان مطلوب، ڈیسک آفیسر، نظامت نصاب و تعلیم اساتذہ خیبر پختونخوا، ایبٹ آباد
شکیل اقبال، سابقہ ماہر مضمون، خیبر پختونخوا ٹیکسٹ بک بورڈ، پشاور

ایڈیٹر: ناصر کمال، ڈیسک آفیسر، خیبر پختونخوا ٹیکسٹ بک بورڈ، پشاور

طباعت زیر نگرانی: ظریف المعانی (چیئرمین)
عبدالولی خان، ممبر (ای اینڈ پی)، خیبر پختونخوا ٹیکسٹ بک بورڈ، پشاور

تعلیمی سال: 2023-24ء
ویب سائٹ: www.kptbb.gov.pk
ای میل: membertbb@yahoo.com
ٹیلی فون نمبر: 091-9217159-60

فہرست عنوانات

باب نمبر	عنوان	صفحہ نمبر
باب 1:	**پروگرامنگ کا طریقہ کار**	1
	1.1 مسئلے کو سمجھنا	2
	1.1.1 مسئلہ	2
	1.1.2 مسئلے کا تجزیہ	3
	1.1.3 مسئلے کے حل کی منصوبہ بندی کرنا	4
	1.1.4 مسئلے کے صریح حل	5
	1.1.5 بہترین حل کا چناؤ کرنا	5
	1.2 الگورتھمز	6
	1.2.1 الگورتھم کیا ہے؟	6
	1.2.2 مسئلے کے حل میں الگورتھم کا کردار	6
	1.2.3 الگورتھم کی کارکردگی کا معیار جانچنے کے پیمانے	6
	1.2.4 مسائل کے حل کے لیے الگورتھم کا استعمال	8
	1.3 فلو چارٹ	13
	1.3.1 فلو چارٹ کیا ہے؟	13
	1.3.2 مسئلے کے حل میں فلو چارٹ کی اہمیت	13
	1.3.3 فلو چارٹ کی ضروریات	14
	1.3.4 فلو چارٹ کی علامات	15
	1.3.5 چند مسائل کے حل کے لیے لکھے گئے فلو چارٹ	16
	اہم نکات	21
	مشق	22
باب نمبر 2:	**C پروگرامنگ**	24
	2.1 پروگرامنگ کا تعارف	25
	2.1.1 کمپیوٹر پروگرام	25

NOT FOR SALE

صفحہ نمبر	عنوان	باب نمبر
25	2.1.2 پروگرامنگ لینگویجز	
29	2.1.3 ہائی لیول لینگویجز کی خصوصیات	
31	2.1.4 چند مقبول ہائی لیول پروگرامنگ لینگویجز	
32	2.1.5 کمپائلر اور انٹرپریٹر	
33	2.2 پروگرامنگ کا انوائرمنٹ	
33	2.2.1 انٹیگریٹڈ ڈیویلپمنٹ انوائرمنٹ (IDE)	
34	2.2.2 C پروگرامنگ ماحول کے مختلف اجزاء	
35	2.3 پروگرامنگ کی اساسیات	
36	2.3.1 ہیڈر فائل	
37	2.3.2 کلیدی الفاظ	
37	2.3.3 C پروگرام کا بنیادی ڈھانچہ	
40	2.3.4 C میں کمنٹس کا استعمال	
41	2.4 متغیرات اور مستقلات	
41	2.4.1 متغیر	
42	2.4.2 مستقل	
42	2.4.3 متغیر کا نام رکھنے کے لیے اصول	
43	2.4.4 C میں استعمال ہونے والی ڈیٹا کے اقسام	
45	2.4.5 ٹائپ کاسٹنگ	
46	2.4.6 const کا استعمال	
47	2.4.7 ویری ایبل بنانا (Declaration) اور ابتدائی قیمت رکھنا (Initialization)	
48	اہم نکات	
49	مشق	
51	باب نمبر 3: اِن پُٹ اور آؤٹ پُٹ کا عمل	
52	3.1 اِن پُٹ آؤٹ پُٹ فنکشنز	
52	3.1.1 آؤٹ پُٹ فنکشن	

NOT FOR SALE

باب نمبر	عنوان	صفحہ نمبر
	3.1.2 اِن پُٹ فنکشن	55
	3.1.3 اسٹیٹمنٹ ٹرمینیٹر (Semicolon)	60
	3.1.4 فارمیٹ سپیسیفائر	60
	3.1.5 اسکیپ سیکوئنسز	62
	3.2 آپریٹرز	63
	3.2.1 اریتھمیٹک حسابی آپریٹر	64
	3.2.2 اسائنمنٹ آپریٹر (=)	65
	3.2.3 کمپاؤنڈ اسائنمنٹ آپریٹرز	65
	3.2.4 انکریمنٹ آپریٹر (++)	66
	3.2.5 ڈیکریمنٹ آپریٹر (--)	68
	3.2.6 نسبتی یا ریلیشنل آپریٹرز	69
	3.2.7 منطقی یا لاجیکل آپریٹر	70
	3.2.8 اسائنمنٹ آپریٹر (=) اور مساوی ہے (==) آپریٹر میں فرق	71
	3.2.9 یونری اور بائنری آپریٹر میں فرق	71
	3.2.10 کنڈیشنل آپریٹر (?:)	72
	3.2.11 آپریٹرز اور انکی فوقیت	72
	اہم نکات	
	مشق	
باب نمبر 4: کنٹرول سٹرکچر		77
	4.1 کنٹرول سٹرکچر	78
	4.1.1 کنٹرول اسٹیٹمنٹس	78
	4.1.2 کنڈیشنل / سیلیکشن اسٹیٹمنٹس	78
	4.1.3 if اسٹیٹمنٹ	79
	4.1.4 if-else اسٹیٹمنٹ	81
	4.1.5 switch اسٹیٹمنٹ	83

NOT FOR SALE

باب نمبر	عنوان	صفحہ نمبر
	4.1.6 نیسٹڈ سیلیکشن سٹرکچر کا استعمال	86
	4.1.7 سیلیکشن اسٹرکچر کے درمیان فرق	88
	4.1.8 یونٹ 1 کے فلوچارٹس کے لئے پروگرام کوڈز	88
	اہم نکات	90
	مشق	91
باب نمبر 5: لُوپ سٹرکچر		93
	5.1 تعارف	94
	5.1.1 لُوپ سٹرکچر	94
	5.1.2 for لُوپ	95
	5.1.3 while لُوپ	97
	5.1.4 do-while لُوپ	99
	5.1.5 break اسٹیٹمنٹ	102
	5.1.6 continue اسٹیٹمنٹ	103
	5.1.7 مختلف لُوپ سٹرکچرز میں فرق	104
	5.1.8 نیسٹڈ لُوپ	104
	5.1.9 باب 1 کے فلوچارٹس کے لئے کوڈز	106
	اہم نکات	108
	مشق	[illegible]
باب نمبر 6: کمپیوٹر لاجک اور لاجک گیٹس		110
	6.1 کمپیوٹر میں ڈیٹا کا اظہار	
	6.2 لاجک گیٹ	
	6.2.1 ڈیجیٹل لاجک اور لاجک گیٹس	111
	6.2.2 بنیادی لاجک گیٹس	112
	6.2.3 ٹروتھ ٹیبل	113
	6.2.4 مزید لاجک گیٹس اور انکے ٹروتھ ٹیبل	116

NOT FOR SALE

صفحہ نمبر	عنوان	باب نمبر
118	6.2.5 بنیادی گیٹس کی مدد سے NAND، NOR، XOR اور XNOR گیٹ بنانا	
121	6.2.6 بُولین ایکسپریشن کو لاجک سرکٹ میں تبدیل کرنا	
122	6.3 بُولین فنکشن کو کارناف میپ کی مدد سے مختصر کرنا	
122	6.3.1 کارناف میپ	
125	6.3.2 دو ویری ایبل والے بولین فنکشن کا K-Map کے ذریعے اختصار	
126	6.3.3 تین ویری ایبل والے بُولین فنکشن کا K-Map کے ذریعے اختصار	
128	6.3.4 مختصر شدہ فنکشن سے لاجک سرکٹ بنانا	
131	اہم نکات	
131	مشق	
133	باب نمبر 7: ورلڈ وائیڈ ویب اور HTML	
134	7.1 ورلڈ وائیڈ ویب کا تعارف	
134	7.1.1 ورلڈ وائیڈ ویب کی اصطلاحات	
141	7.1.2 ویب سائٹس کی مختلف اقسام	
146	7.2 HTML کا تعارف	
146	7.2.1 ہائپر ٹیکسٹ مارک اپ لینگویج	
146	7.2.2 HTML ڈاکومنٹ بنانا اور براؤزر میں دکھانا	
148	7.2.3 HTML ٹیگز کا استعمال	
148	7.2.4 HTML کے ہیڈ اور باڈی ٹیگز	
149	7.3 ٹیکسٹ فارمیٹنگ	
149	7.3.1 ٹیکسٹ فارمیٹنگ کی اساسیات	
151	7.3.2 ٹیکسٹ فارمیٹنگ ٹیگز اور ان کا استعمال	
155	7.4 HTML میں فہرست/لسٹ بنانا	
156	7.4.1 لسٹ کی مختلف اقسام	
157	7.4.2 فہرست/لسٹ بنانا	
161	7.5 تصاویر اور پس منظر	
161	7.5.1 ویب پیج میں تصویر لگانا	
163	7.5.2 ویب پیج میں تصویر کا سائز مقرر کرنا	

NOT FOR SALE

صفحہ نمبر	عنوان	باب نمبر

صفحہ نمبر	عنوان	باب نمبر
165	7.5.3 پس منظر اور پیش منظر کا رنگ بدلنا	
166	7.5.4 پس منظر میں تصویر لگانا	
167	7.6 ہائپر لنک	
167	7.6.1 ہائپر لنک کیا ہے؟	
167	7.6.2 اینکر ٹیگ	
167	7.6.3 ویب پیج کی طرف ہائپر لنک بنانا	
168	7.6.4 ویب پیج کے اندر ہائپر لنک بنانا	
171	7.6.5 گرافیکل ہائپر لنک تیار کرنا	
172	7.7 جدول یا ٹیبل بنانا	
177	7.8 فریم بنانا	
177	7.8.1 فریم	
177	7.8.2 فریم سیٹ	
177	7.8.3 فریم سیٹ تیار کرنا	
178	7.8.4 ایک سے زیادہ فریمز کے ساتھ فریم سیٹ بنانا	
180	اہم نکات	
181	مشق	
182	سوالات کے جوابات	
183	فرہنگ	
188	انڈیکس	
191	ضمیمہ نمبر 1: اشکالات کی فہرست	
196	ضمیمہ نمبر 2: جدول کی فہرست	
197	ضمیمہ نمبر 3: نشانات کی فہرست	
198	ضمیمہ نمبر 4: Notepad ++ کی کمانڈز کی فہرست	
199	ضمیمہ نمبر 5: کتابیات	

NOT FOR SALE

باب نمبر 1

پروگرامنگ کا

اس باب کی تکمیل کے بعد طلبہ اس قابل ہو جائیں گے کہ وہ:

- مسئلے کی تعریف اور اس کا تجزیہ کر سکیں۔
- مسئلہ کے حل کے لیے منصوبہ بندی کر سکیں۔
- مسئلے کے تمام درست حل واضح کر کے ان میں سے بہترین حل کا انتخاب کر سکیں۔
- الگورتھم کی تعریف کر سکیں۔
- مسئلے کے حل میں الگورتھم کا کردار بیان کر سکیں۔
- الگورتھم کی کارکردگی جانچنے کے اصول بیان کر سکیں۔
- مختلف حسابی سوالات کے حل کے لیے الگورتھم لکھ سکیں۔
- فلوچارٹ کی تعریف کر سکیں۔
- فلوچارٹ کی اہمیت تفصیل سے بیان کر سکیں۔
- کسی بھی حل طلب مسئلے کی ضروریات کا تعین کر سکیں۔
- فلوچارٹ کی علامات کے استعمال کو سمجھ سکیں۔
- مختلف حسابی سوالات کے حل کے لیے فلوچارٹ واضح کر سکیں۔

NOT FOR SALE

تعارف (Introduction)

اس باب میں دو اہم نکات زیر بحث رہیں گے۔ سب سے پہلے ہم کمپیوٹر سائنس میں الگورتھم اور ڈیٹا سٹرکچر کے بر محل استعمال کا جائزہ لیں گے۔ خاص طور پر یہ کہ ہمیں ان دونوں موضوعات کو کیوں پڑھنا چاہیے اور ان کو جاننے کے بعد ہم کس طرح ایک اچھے حل کنندہ (Problem Solver) بن سکتے ہیں۔ اس کے بعد ہم فلو چارٹ کے تصورات کا جائزہ لیں گے۔

1.1 مسئلے کو سمجھنا (Understanding a Problem)

کمپیوٹر کے ذریعے کسی بھی مسئلے کو حل کرنے کے لیے ہمیں ایک کمپیوٹر پروگرام لکھنا پڑتا ہے۔ کمپیوٹر پروگرام لکھنے کے لیے ہمیں دو چیزیں معلوم ہونی چاہئیں؛ آپ کمپیوٹر سے کیا کروانا چاہتے ہیں اور وہ کونسے ذرائع ہیں جن کی مدد سے آپ کمپیوٹر سے کام لینا چاہ رہے ہیں۔ پہلا حصہ (جسے آپ الگورتھم بنانا (Algorithm Development) کہہ سکتے ہیں) دوسرے حصے کے مقابلے میں زیادہ مشکل ہے۔ اس مقصد کے لیے بہت سارے آلات جیسے کہ فلو چارٹ اور سوڈوکوڈ (Pseudo Code) موجود ہیں جن سے اس عمل میں مدد لی جا سکتی ہے۔

شکل نمبر 1.1 مسئلے کو سمجھنا

1.1.1 مسئلہ (The Problem)

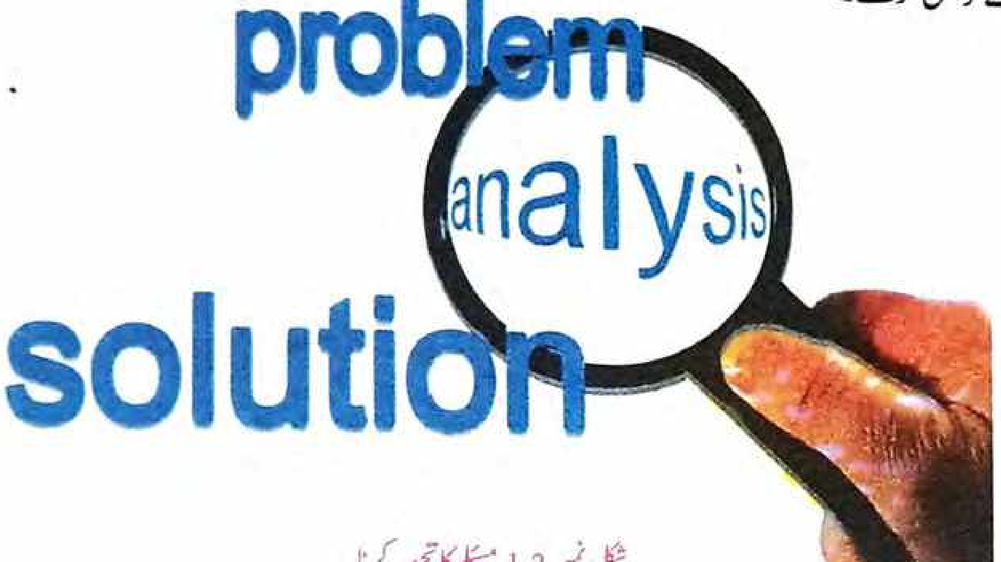
مسئلہ ایک رکاوٹ، مشکل یا چیلنج، یا کوئی بھی ایسی صورت حال ہے جس کو حل کرنا مقصود ہو۔ مسئلہ کی سب سے پہلی پہچان یہ ہے کہ جو ہونا چاہیئے اور جو ہو رہا ہے اس میں فرق آجائے۔ مسئلے کی آسان تعریف کرنے کے لیے اپنے آپ سے چند سوالات پوچھیں۔ مثال کے طور پر اگر آپ ایک ایسا پروگرام لکھنا چاہتے ہیں جس میں آپ پانچ نمبروں کا اوسط (Average) معلوم کرنا چاہتے ہیں تو اپنے آپ سے چند سوالات پوچھیں:

NOT FOR SALE

- اوسط کا اصل مطلب کیا ہے؟
- آپ کیسے اوسط معلوم کر سکتے ہیں؟
- اس طرح سے ہم مسئلے یا مشکل کو درست طور پر بیان کر سکتے ہیں۔

1.1.2 مسئلے کا تجزیہ (Problem Analysis)

کمپیوٹر بذات خود اتنا سمجھدار نہیں ہے کہ وہ کسی بھی مسئلے کا تجزیہ کر سکے اور پھر اس کے لیے حل بھی تجویز کر سکے۔ یہ صرف انسان (یعنی پروگرامر) کا کام ہے کہ وہ کسی بھی مسئلے کا تجزیہ کرے اور اس مسئلے کو حل کرنے کے لیے ہدایات (Code) لکھے اور پھر یہ ہدایات کمپیوٹر کو مہیا کر کے مسئلے کو حل کرے۔



شکل نمبر 1.2 مسئلے کا تجزیہ کرنا

مسئلے کے چناؤ اور وضاحت ہو جانے کے بعد سب سے پہلے جس چیز کی ضرورت ہے وہ یہ ہے کہ مسئلے کی وجوہات کا تعین کیا جائے۔

وجوہات کا تجزیہ کرنے کے لیے:

- مسئلے کے متعلق آپ کا تصور بالکل واضح ہونا چاہیے تاکہ آپ اس کا درست تجزیہ کر سکیں۔
- مسئلے کے متعلق ابھرنے والے تمام سوالات پر ایک دفعہ پھر نظر ثانی کر لیں تاکہ یہ اطمینان ہو جائے کہ آپ کے سامنے درج ذیل نکات واضح ہیں:
  - مسئلہ کس سے متعلق ہے؟
  - مسئلے سے کتنی دفعہ پالا پڑتا ہے؟
  - یہ کب اور کہاں پایا جاتا ہے؟
  - اس سے کون متاثر ہوتا ہے؟
- ممکنہ وجوہات کے بارے میں سوچیں۔
- وجوہات کی گروپ بندی کریں۔
- وجوہات کی تصویروں سے وضاحت کریں۔

جب مسئلے کی وجوہات کا تجزیہ ہو جائے تو پھر ان کے حل کے لیے منصوبہ بندی کریں۔

NOT FOR SALE

1.1.3 مسئلے کے حل کی منصوبہ بندی (Planning a Solution)

تجزیے کے بعد، مسئلے کے حل کی منصوبہ بندی کی جاتی ہے۔ کسی بھی مسئلے کو متبادل طریقوں سے حل کیا جا سکتا ہے۔ تاہم بہترین حل اسے کہا جائے گا جو کم لاگت، کم وقتی، تیز اور کم پیچیدہ ہو۔

شکل نمبر 1.3 مسئلے کے حل کے لیے منصوبہ بندی

مسئلے کے حل کے لیے ذیل میں چند طریقہ کار بیان کیے گئے ہیں:

- **تجرید Abstraction**: کسی بھی عملی صورتحال پر ممکنہ حل کو لاگو کرنے سے پہلے اس کو کسی بھی نمونہ (Model) پر آزمایا جائے۔
- **قیاس Analogy**: اُس حل کو آزمایا جائے جو اس مسئلے سے ملتے جلتے مسئلے کو حل کر چکا ہو۔
- **دماغ لڑانا Brainstorming**: لوگوں کے ایک گروہ میں تمام ممکنہ حل پیش کیے جائیں تاکہ ان پر بحث و مباحثہ کے بعد بہترین حل تیار کیا جا سکے اور اس کا چناؤ کیا جا سکے۔

شکل نمبر 1.4 ممکنہ جوابات (Brainstorming)

NOT FOR SALE

- **تقسیم اور فتح Divide and Conquer**: بڑے اور پیچیدہ مسئلے کو چھوٹے چھوٹے آسان قابل حل مسئلوں میں تقسیم کیا جائے تاکہ حل کرنا آسان ہو جائے۔
- **اختصار Reduction**: مسئلے کو دوسرے ایسے مسائل میں تبدیل کیا جائے جن کا حل پہلے سے موجود ہو۔
- **تحقیق Research**: اپنے تجربات کی روشنی میں، ایک جیسے مسائل کے پہلے سے موجود حل کو اپنے مقصد کے لیے تبدیل کریں۔
- **سعی و خطا Trial and Error**: تمام ممکنہ حل آزمانا جب تک درست حل نہ مل جائے۔

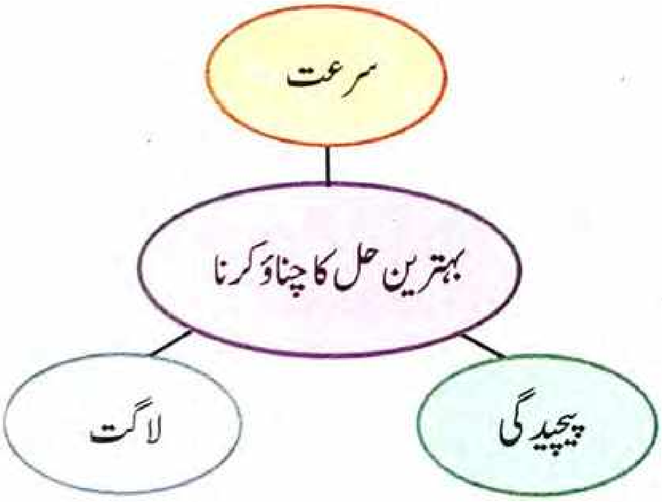
1.1.4 مسئلے کے صریح حل (Candid Solutions of a Problem)

"مسئلے کو حل کرنے کے عمل کو" حل ہی کہا جائے گا۔ مسئلے کا بہترین اور پسندیدہ حل وہ ہوگا جس کی لاگت کم سے کم ہو۔ مسئلے کو حل کرنے کے لیے بہت سے متبادل طریقے ہو سکتے ہیں۔ تاہم مسئلے کا بہترین اور مناسب ترین حل معلوم کرنا ہی مسئلے کا صریح حل قرار دیا جائے گا۔

1.1.5 بہترین حل کا چناؤ کرنا (Selecting the best Solution)

کسی بھی مسئلے کے بہترین حل کے چناؤ کے لیے درج ذیل عناصر بہت اہمیت کے حامل ہیں۔

- سرعت
- لاگت
- پیچیدگی



شکل نمبر 1.5 بہترین حل کا چناؤ کرنا

i. سرعت (Speed)

مسئلے کے لیے تجویز شدہ حل کتنی تیزی سے مسئلے کو حل کر سکتا ہے یہ ایک ایسا پیمانہ ہے جو مؤثر ترین حل کی نشاندہی کرتا ہے۔

ii. لاگت (Cost)

کسی حل کی لاگت سے مراد تجویز شدہ حل کے لیے درکار میموری، ہارڈ ویئر اور سافٹ ویئر ہیں۔ کسی بھی حل کو اس وقت بہترین کہا جائے گا جو کم سے کم لاگت میں مناسب ترین اور مطلوبہ نتائج مہیا کرے۔

NOT FOR SALE

iii. پیچیدگی (Complexity)

کسی بھی حل کے لیے درکار وسائل خاص طور پر، درکار کُل وقت، اس مسئلے کی پیچیدگی کو ظاہر کرتا ہے۔ یعنی کتنے مراحل سے کتنی میموری استعمال کر کے حل تک پہنچنا ہے۔

1.2 الگورتھمز (Algorithms)

1.2.1 الگورتھم کیا ہے؟ (What is an Algorithm)

مناسب اور منطقی اقدامات کی وہ فہرست جو کسی مخصوص مسئلے کے حل کے طور پر لکھی جاتی ہیں انہیں الگورتھم کہا جاتا ہے۔ غیر رسمی طور پر الگورتھم کی تعریف اس طرح بھی کی جاسکتی ہے کہ الگورتھم ایک ایسا مرحلہ وار حسابی عمل ہے جو ایک عدد یا رقم یا اعداد کے مجموعے کو بطور ان پُٹ لیتا ہے اور کوئی بھی رقم یا عدد یا اعداد کے مجموعہ کو نتیجہ کے طور پر آؤٹ پٹ میں ظاہر کرتا ہے۔ یعنی الگورتھم حسابی مرحلوں / اقدامات کا ایک ترتیب وار سلسلہ ہے جو اِن پُٹ کو آؤٹ پُٹ میں تبدیل کرتا ہے۔ پس الگورتھم کو کسی بھی مسئلے کے حل کے طور پر بھی استعمال کیا جاسکتا ہے۔

1.2.2 مسئلے کے حل میں الگورتھم کا کردار (Role of Algorithm in Problem Solving)

کسی بھی مسئلے کو حل کرنے کے لیے الگورتھم بہت اہم کردار ادا کرتا ہے۔ مثال کے طور پر اگر کوئی پیچیدہ مسئلہ درپیش ہو تو اس کے لیے بہت ساری معلومات جمع کرنا پڑیں گی اور بہت سارے ڈیٹا کو پروسیس کرنا پڑے گا۔ لیکن اگر ہم معلومات اکٹھا کرنے اور ڈیٹا پروسیس کرنے کے اس عمل کو چھوٹے چھوٹے ترتیب وار مراحل میں تقسیم کر دیں تو اس سے ہمارا پیچیدہ مسئلہ حل کرنا کافی حد تک آسان ہو جائے گا۔ پیچیدہ مسائل کے حل کے لئے تقسیم اور فتح (Divide and Conquer) ایک بہت ہی مشہور اور کار آمد اصول ہے جس سے مسائل کا حل پیش کرنا بہت آسان ہو جاتا ہے۔ مثال کے طور پر 20 نمبروں کا اوسط معلوم کرنا ہو تو اس مسئلے کو اگر چھوٹے چھوٹے مجرد (Discrete) اقدامات کے مجموعے یعنی الگورتھم میں تقسیم کیا جائے تو اس سے مسئلے کو آسانی سے حل کیا جاسکتا ہے۔

1. کوئی سے 20 اعداد منتخب کریں۔
2. تمام اعداد کو جمع کریں۔
3. مرحلہ نمبر 2 سے حاصل شدہ نتیجہ کو 20 سے یعنی اعداد کی کل تعداد سے تقسیم کریں۔
4. اوسط نکل آئے گا۔

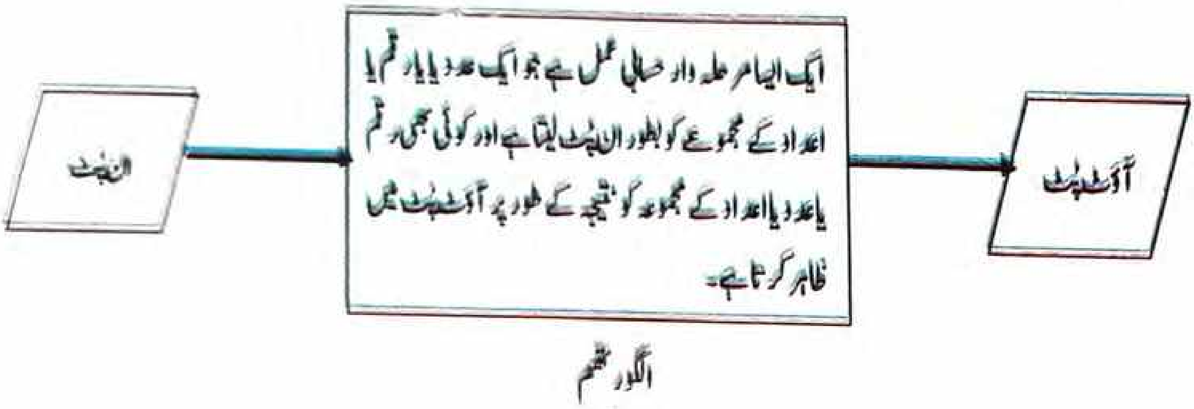
1.2.3 الگورتھم کی کارکردگی کا معیار جانچنے کے پیمانے

(Criteria for Measuring Efficiency of an Algorithm)

کمپیوٹر سائنس میں کسی بھی الگورتھم کی خصوصیات کو پرکھ کر ہم اس کی کارکردگی معلوم کر سکتے ہیں۔ الگورتھم کی خصوصیات میں ہم یہ جانچنے کی کوشش کرتے ہیں کہ کوئی بھی الگورتھم کس حد تک وسائل کو استعمال کرتا ہے؟ بہت ہی عام استعمال ہونے والے دو پیمانے درج ذیل ہیں:

- سرعت یا چلنے کے لیے درکار وقت (Speed or Running Time): یہ وہ وقت ہے جو ایک الگورتھم مکمل ہونے کے لیے لیتا ہے۔
- درکار میموری: کسی آپریشن کے دوران الگورتھم کی طرف سے استعمال کی گئی میموری۔

مزید بر آں اسکی کارکردگی کو درج ذیل عوامل کی بنیاد پر ماپا جاسکتا ہے۔



شکل نمبر 1.6 الگورتھم کی کارکردگی

i. درکار ان پٹ (Inputs needed)

کسی الگورتھم کو ملنے والی ان پٹ کی مقدار سے پتا لگایا جاسکتا ہے کہ اس الگورتھم کو کتنا وقت درکار ہے۔
مثال کے طور پر ایک چھوٹے قالب (Matrix) کے مقابلے میں ایک بڑے قالب کو ضرب دینے کے لیے زیادہ وقت درکار ہو گا۔ اسی طرح کبھی کبھار کوئی الگورتھم ایک جیسی مقدار کی ان پٹ کے لئے مختلف وقت صرف کرتا ہے۔

ii. پروسیسنگ کو مکمل کرنا (Processing to be completed)

الگورتھم کو اپنی پروسیسنگ مکمل کرنے کے لیے کتنا وقت درکار ہے اس سے الگورتھم کی کارکردگی کو ماپا جاسکتا ہے۔ پروسیسنگ کی تکمیل کا وقت سائیکل ٹائم، انتظار (Wait) اور پروسیسنگ کا وقت (Processing Time) پر مشتمل ہوتا ہے۔

iii. فیصلہ کرنا (Decision to be taken)

مسئلے کو حل کرنے کے دوران مختلف فیصلے کرنا ایک بہت ہی اہم صلاحیت ہے جس سے الگورتھم کی کارکردگی میں بہتری لائی جاسکتی ہے اور موزوں ترین نتائج حاصل کیئے جاسکتے ہیں۔

iv. حاصل شدہ آؤٹ پٹ (Outputs to be provided)

اگر ایک الگورتھم اپنی ہر ان پٹ کے لیے مطلوبہ نتائج مہیا کرے تو اس کا مطلب یہ ہے کہ یہ الگورتھم درست ہے۔ کیونکہ ایک غلط الگورتھم کچھ ان پٹس پر ختم ہی نہیں ہو گا یا وہ ایسے نتائج فراہم کرے گا جو کہ ناقابل قبول ہوں گے۔ اس لیے الگورتھم والے حل جو قابل قبول نتائج فراہم کریں ان کو ہی مسئلے کا بہترین حل کہا جائے گا۔

NOT FOR SALE

1.2.4 مسائل کے حل کے لیے الگورتھم کا استعمال (Algorithm for Solving Some Problems)

مسئلہ نمبر 1

پانچ نمبروں کا حاصل جمع، حاصل ضرب اور اوسط معلوم کرنا۔

الگورتھم: فرض کیا کہ پانچ اعداد دیئے گئے ہیں n1, n2, n3, n4, n5۔ یہ الگورتھم ان اعداد کا حاصل ضرب PRO، حاصل جمع SUM اور اوسط AVG معلوم کرے گا۔

```
Step 1.   Read n1, n2, n3, n4, n5
Step 2.   Set SUM =n1+n2+n3+n4+n5
Step 3.   Set PRO =n1*n2*n3*n4*n5
Step 4.   Set AVG =SUM/5
Step 5.   Write SUM, PRO, AVG
Step 6.   Exit
```

مسئلہ نمبر 2

دیئے گئے تین غیر مساوی اعداد میں سے سب سے بڑے عدد کو منتخب کرنا۔

الگورتھم: فرض کیا تین اعداد n1, n2, n3 ہیں۔ یہ الگورتھم ان تین اعداد میں سے سب سے بڑا عدد (LARG) معلوم کرے گا۔

```
Step 1.   Read n1, n2, n3
Step 2.   If n1>n2 AND n1>n3 then Set LARG=n1
          Else if n2>n3 then Set LARG=n2
          Else Set LARG=n3
          [End of If structure]
Step 3.   Write LARG
Step 4.   Exit
```

مسئلہ نمبر 3

دی گئی قوت اور کمیت کی بنیاد پر کسی بھی متحرک جسم کا اسراع معلوم کرنا۔

الگورتھم: یہ الگورتھم کسی بھی متحرک چیز کا اسراع ACCEL معلوم کرے گا جبکہ اس چیز کی کمیت MASS اور لگائی گئی قوت FORCE دی گئی ہو۔

```
Step 1.   Read MASS, FORCE
Step 2.   Set ACCEL= FORCE / MASS
Step 3.   Write ACCEL
Step 4.   Exit
```

مسئلہ نمبر 4

کسی کُرے (Sphere) اور مکعب (Cube) کا حجم معلوم کرنا۔

الگورتھم: یہ الگورتھم کسی مکعب کا حجم VC جس کے ایک ضلع کی لمبائی A ہو اور کُرے کا حجم VS جس کا رداس (Radius) R ہو، معلوم کرے گا۔

NOT FOR SALE

Step 1. Read A, R.
Step 2. Set VC= A*A*A
Step 3. Set VS= 4/3 * 3.14*R*R*R
Step 4. Write VC, VS
Step 5. Exit

مسئلہ نمبر 5

مثلث کا رقبہ معلوم کرنا۔

الگورتھم: یہ الگورتھم کسی بھی مثلث، جس کے قاعدہ کی لمبائی B اور اونچائی H ہو، کا رقبہ AREA معلوم کرتا ہے۔

Step 1. Read B, H
Step 2. Set AREA= B*H / 2
Step 3. Write AREA
Step 4. Exit

مسئلہ نمبر 6

متوازی الاضلاع (Parallelogram) کا رقبہ معلوم کرنا۔

الگورتھم: یہ الگورتھم ایسے متوازی الاضلاع کا رقبہ AP معلوم کرے گا جس کے قاعدہ کی لمبائی B اور اونچائی H ہو۔

Step 1. Read B, H
Step 2. Set AP= B*H
Step 3. Write AP
Step 4. Exit

مسئلہ نمبر 7

ذوزنقہ (Trapezium) کا رقبہ معلوم کرنا۔

الگورتھم: یہ الگورتھم کسی ذوزنقہ کا رقبہ ATRAP معلوم کرتا ہے، جس کے متوازی اضلاع کی لمبائی A اور B ہو اونچائی H (اضلاع کے درمیان عمودی فاصلہ) ہو۔

Step 1. Read A, B
Step 2. Set ATRAP $= \frac{(A+B)}{2} \times H$
Step 3. Write ATRAP
Step 4. Exit

مسئلہ نمبر 8

مضامین میں حاصل شدہ نمبروں کی بنیاد پر گریڈ کا تعین کرنا۔

الگورتھم: یہ الگورتھم مضامین میں حاصل شدہ نمبروں SUBMARKS کی بنیاد پر ان مضامین کے لیے گریڈ GRADE کا تعین کرے گا۔

Step 1. Read SUBMARKS
Step 2. If SUBMARKS>=80 Set GRADE= A+
Else if SUBMARKS>=70 Set GRADE= A
Else if SUBMARKS>=60 Set GRADE= B
Else if SUBMARKS>=50 Set GRADE= C

NOT FOR SALE

Else if SUBMARKS>=40 Set GRADE= D
Else if SUBMARKS>=33 Set GRADE= E
Else Set GRADE=FAIL
[End of if structure]

Step 3. Write GRADE
Step 4. Exit

مسئلہ نمبر 9

کئی سال بعد متوقع آبادی معلوم کرنا۔

الگورتھم: یہ الگورتھم موجود آبادی P اور نمو کی شرح R رکھنے والے علاقے کی T سال بعد متوقع آبادی PE کو معلوم کرتا ہے۔

Step 1. Read P, R, T
Step 2. Set EP= P*R*T
Step 3. Write EP
Step 4. Exit

مسئلہ نمبر 10

دیئے گئے عدد کی طاقت معلوم کرنا۔

الگورتھم: یہ الگورتھم کسی بھی دیئے گئے عدد NUM کی X طاقت EXP معلوم کرتا ہے۔

Step 1. Read NUM, X
Step 2. Set EXP= NUM ^ X (^ کیرٹ قوت نما کے لیے استعمال ہوتا ہے)
Step 3. Write EXP
Step 4. Exit

مسئلہ نمبر 11

دیئے گئے دو اعداد کے درمیان پے در پے طاق اعداد معلوم کرنا۔

الگورتھم: یہ الگورتھم کسی عدد NUM سے لے کر آخری عدد N تک مسلسل طاق اعداد کو معلوم کرتا ہے۔

Step 1. Read NUM, N
Step 2. Repeat Steps 3 and 4 while NUM<= N
Step 3. Write NUM
Step 4. Set NUM=NUM+2
[End of while loop]
Step 5. Exit

مسئلہ نمبر 12

کسی بھی دیئے ہوئے عدد کے لیے دو اعداد کے درمیان جزو ضربی (Multiples) کی کل تعداد معلوم کرنا۔

الگورتھم: یہ الگورتھم کسی عدد NUM کے لیے دو اعداد N1 اور N2 کے درمیان آنے والے اجزائے ضربی کی کل تعداد معلوم کرتا ہے

جبکہ N2 بڑا ہے N1 سے۔

NOT FOR SALE

Step 1. Read NUM, N1, N2
Step 2. Set M = 0 [Initialize M.]
Step 3. For A=N1 to N2
Step 4. Set MUL= NUM % A (% is used for remainder)
Step 5. If MUL = 0 Then M=M+1
Step 6. Next A
Step 7. Write M
Step 8. Exit

مسئلہ نمبر 13

کسی بھی دیئے گئے عدد کے لیے ضرب کا پہاڑہ بنانا۔

الگورتھم: یہ الگورتھم دیئے گئے عدد NUM کے لیے پہاڑہ بناتا ہے۔

Step 1. Set C=1 [Initialize counter.]
Step 2. Read NUM
Step 3. Repeat Steps 4,5 and 6 while C<=10
Step 4. Set TAB= NUM*C
Step 5. Write NUM '*' C '=' TAB
Step 6. C=C+1 [Increment counter variable.]
[End of while loop]
Step 7. Exit

مسئلہ نمبر 14

سینٹی گریڈ درجہ حرارت کو فارن ہائیٹ درجہ حرارت میں تبدیل کرنا۔

الگورتھم: یہ الگورتھم سینٹی گریڈ درجہ حرارت C کو فارن ہائیٹ F درجہ حرارت میں تبدیل کرتا ہے۔

Step 1. Read C
Step 2. Set F= 32+C*9/5
Step 3. Write F
Step 4. Exit

مسئلہ نمبر 15

فارن ہائیٹ درجہ حرارت کو سینٹی گریڈ درجہ حرارت میں تبدیل کرنا۔

الگورتھم: یہ الگورتھم فارن ہائیٹ درجہ حرارت F کو سینٹی گریڈ C درجہ حرارت میں تبدیل کرتا ہے۔

Step 1. Read F
Step 2. Set C= (F-32)*5/9
Step 3. Write C
Step 4. Exit

مسئلہ نمبر 16

دو صحیح اعداد کے درمیان تمام جفت اعداد معلوم کرنا (جبکہ n2 بڑا ہے n1 سے)۔

الگورتھم: یہ الگورتھم دو اعداد n1 اور n2 کے درمیان آنے والے تمام جفت اعداد معلوم کرتا ہے جبکہ n1 چھوٹا ہے n2 سے اور n1 ایک جفت عدد ہے۔

NOT FOR SALE

Step 1. Read n1, n2
Step 2. Set EVEN=n1
Step 3. Repeat Steps 4 and 5 while EVEN<= n2
Step 4. Write EVEN
Step 5. Set EVEN=EVEN +2
[End of while loop]
Step 6. Exit

مسئلہ نمبر 17

اعداد کی دو علیحدہ فہرستوں میں موجود نمبروں کو جمع کرنا۔

الگورتھم: یہ الگورتھم دو فہرستوں L1[n] اور L2[n] میں موجود متعلقہ اعداد کو بالترتیب جمع کر کے ایک تیسری فہرست L3[n] میں رکھتا ہے۔

Step 1. Read L1[n], L2[n]
Step 2. Set I=1 [Initialize counter.]
Step 3. Repeat Steps 4 and 5 while I<= n
Step 4. Set L3[I]=L1[I]+L2[I]
Step 5. Set I=I+1 [Increment counter.]
[End of while loop]
Step 6. Exit

مسئلہ نمبر 18

دو اعداد کا عاد اعظم (GCD) معلوم کرنا۔

الگورتھم: یہ الگورتھم دیئے گئے دو اعداد M اور N کا عاد اعظم (GCD) معلوم کرتا ہے۔

Step 1. Read M, N
Step 2. If N=0, Go to Step 5; otherwise, Proceed to Step 3
Step 3. Divide M by N and assign the value of the remainder to R
Step 4. Assign the value of N to M and the value of R to N and go to Step 2
Step 5. Write M
Step 6. Exit

مسئلہ نمبر 19

معلوم کرنا کہ ایک دیا گیا عدد مفرد عدد (Prime Number) ہے یا نہیں۔

الگورتھم: یہ الگورتھم معلوم کرے گا کہ ایک دیا گیا عدد NUM مفرد عدد (Prime Number) ہے یا نہیں۔

Step 1. Read NUM
Step 2. Set i=2 [Initialize counter]
Step 3. Repeat step 4 and 5 while i<=NUM/2
Step 4. If NUM%i=0 Then Write 'Not a prime number' and go to Step 7
Step 5. Set i=i+1 [Increment counter.]
[End of while loop]
Step 6. If i=NUM/2, Then Write 'Prime number'
Step 7. Exit

NOT FOR SALE

مسئلہ نمبر 20

دو اعداد n1 اور n2 کے درمیان بالترتیب پائے جانے والے مفرد اعداد (Prime Numbers) معلوم کرنا (جبکہ n1 بڑا ہے n2 سے)

الگورتھم: یہ الگورتھم دو اعداد n1 اور n2 کے درمیان بالترتیب پائے جانے والے مفرد اعداد معلوم کرے گا۔ (جبکہ n1 چھوٹا ہے n2 سے)۔

```
Step 1.   Read n1, n2
Step 2.   Set n=n2+1
Step 3.   Repeat Step 4 while n<n1
Step 4.   Repeat Step 5 for i=2 to  n-1.
Step 5.   If n%i=0 Then Set n=n+1 Go to Step 3
          Else
            (a)Write n
            (b) Set n=n+1
Step 6.   Exit
```

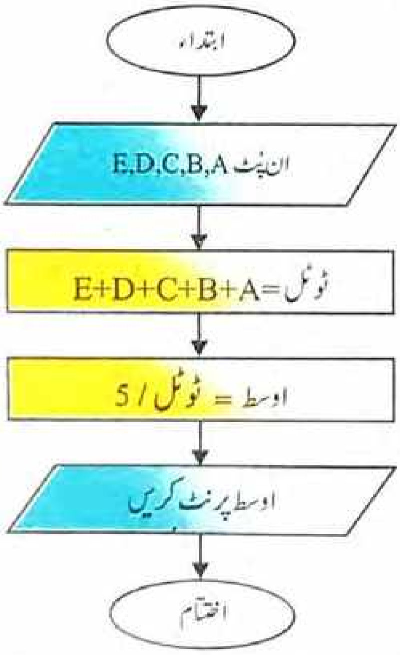
1.3 فلو چارٹ (FLOWCHART)

1.3.1 فلو چارٹ کیا ہے؟ (What is a Flowchart?)

فلو چارٹ دو الفاظ کا مجموعہ ہے جیسے فلو اور چارٹ۔ فلو چارٹ الگورتھم کا تصویری (Diagrammatic) اظہار ہے۔ فلو چارٹ مختلف علامات پر مشتمل ہوتا ہے جو کسی بھی پروگرام کے متعلق معلومات دکھاتا ہے۔ یہ تمام علامات ایک دوسرے سے تیر کے نشان کی مدد سے ملے ہوتے ہیں جو کہ اس عمل کے بہاؤ کی سمت کو ظاہر کرتے ہیں جو آخر میں کسی حل پر منتج ہوتا ہے۔ جیسا کہ شکل نمبر 1.7 میں دکھایا گیا ہے۔

1.3.2 ایک مسئلے کے حل میں فلو چارٹ کی اہمیت

(Importance of a Flowchart for Solving a Problem)



شکل نمبر 1.7 پانچ نمبروں کا اوسط معلوم کرنے کا فلو چارٹ

NOT FOR SALE

کسی مسئلے کو حل کرنے کے لیے فلو چارٹ بہت اہم ہوتے ہیں کیونکہ:

- یہ دیکھنے والے کو پروسیس سمجھنے میں مدد دیتے ہیں۔
- مطلوبہ اِن پُٹ اور آؤٹ پُٹ کو ظاہر کر رہے ہوتے ہیں۔
- پروسیس کے مختلف مراحل کے درمیان تعلق کو ظاہر کر رہے ہوتے ہیں۔
- فیصلہ سازی میں مدد فراہم کرتے ہیں۔
- پروسیس کے بہاؤ کو ظاہر کر رہے ہوتے ہیں۔
- اہم اقدامات کو اُجاگر کرتے ہیں اور غیر اہم مراحل کو ختم کر دیتے ہیں۔

1.3.3 فلوچارٹ کی ضروریات (Flowchart Requirements)

فلوچارٹ بنانے والے کو دیئے گئے مسئلے یا الگورتھم میں موجود درج ذیل ضروریات کو ہر حال میں معلوم کر لینا چاہیے۔

- تمام درکار اِن پُٹ
- پروسیسنگ کی تکمیل
- کیے جانے والے فیصلے
- مہیا کردہ آؤٹ پُٹ

تمام درکار اِن پُٹ (Inputs needed)

ان پُٹ ہر وہ چیز ہے جو پروسیسنگ کے دوران درکار ہوتی ہے۔ فلوچارٹ بنانے والے (Developer) کو پتہ ہونا چاہیے کہ کیا ان پُٹ ہے؟ مسئلے کی تعریف (Problem Statement) سے ان پُٹ معلوم کی جاسکتی ہے۔

مثال کے طور پر اگر کسی دائرے کا رقبہ معلوم کرنا مقصود ہو تو اس مسئلے کے لیے درکار ان پُٹ دائرے کا رداس (R) ہے۔

پروسیسنگ کی تکمیل (Processing to be completed)

جیسا کہ فلوچارٹ کسی نہ کسی کلیے یا کلیات کے مجموعے کا حل پیش کرتے ہیں یا کسی متغیر (Variable) کے لیے ابتدائی قیمت متعین (Initialize) کرتے ہیں۔ اس لیے فلوچارٹ بنانے والے کو پتہ ہونا چاہیے کہ اسے مطلوبہ نتائج حاصل کرنے کے لیے کونسا کلیہ Formula یا حسابی عمل درکار ہے۔ مثال کے طور پر کسی بھی دائرے کا رقبہ معلوم کرنے کے لیے درج ذیل کلیہ استعمال ہوگا۔

Area = Pi * R * R

جبکہ Pi= 3.1415

فیصلہ سازی کرنا (Decision to be taken)

فلوچارٹ میں فیصلہ سازی تب کی جاتی ہے جب کوئی کلیہ یا اظہار خیال صحیح (True) یا غلط (False) ہو۔ مثال کے طور پر جب کسی طالب علم کا نتیجہ معلوم کرنا ہو کہ وہ فیل ہے یا پاس تو اس کے لیے طالب علم کے حاصل کردہ فیصد (PM) پر فیصلہ کیا جائے گا۔ جیسا کہ

If PM>=40 Then Result="PASS" otherwise Result ="FAIL"

NOT FOR SALE

فیصلہ سازی کرنا (Decision to be taken)

آؤٹ پُٹ ہی اصلی نتیجہ ہے۔ فلوچارٹ کے آخر میں آؤٹ پُٹ ضرور دکھانا چاہیے۔ مثال کے طور پر رقبہ، نتیجہ وغیرہ۔

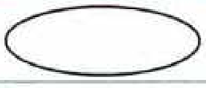
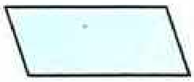
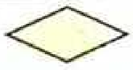
1.3.4 فلوچارٹ کی علامات (Flowchart Symbols)

الگورتھم میں موجود مختلف اقدامات کے لیے فلوچارٹ کے اندر مخصوص شکلیں استعمال ہوتی ہیں ان کو فلوچارٹ کی علامات کہا جاتا ہے۔ لائنوں اور تیر کے نشانات ان اقدامات کی ترتیب اور آپس میں تعلق کو ظاہر کرتے ہیں۔ ذیل میں فلوچارٹ کی چند اہم علامات دی گئی ہیں۔

1. بیضوی **(Oval)**: یہ فلوچارٹ کے آغاز (Start) یا اختتام (End) کو ظاہر کرتا ہے۔
2. تیر کے نشانات **(Arrow Line)**: یہ ایک مرحلے سے دوسرے مرحلے تک کنٹرول کی تبدیلی کو ظاہر کرتے ہیں۔ یہ ایک مرحلے سے دوسرے مرحلے تک پیش رفت کو بھی ظاہر کرتے ہیں۔
3. متوازی الاضلاع **(Parallelogram)**: یہ ان پُٹ اور آؤٹ کو ظاہر کرتے ہیں۔ مثال کے طور پر READ A ، INPUT N ، PRINT N ، DISPLAY B وغیرہ۔
4. مستطیل **(Rectangle)**: یہ پراسیسنگ کے مرحلے کو ظاہر کرتا ہے۔
5. ڈائمنڈ **(Diamond)**: فلوچارٹ میں جب کوئی فیصلہ کرنا ہو یا کسی سوال کا جواب دینا ہو تو ایسی صورتِ حال میں یہ علامت استعمال ہوتی ہے۔ جیسا کہ ہاں یا نہیں (Yes / No) یا صحیح یا غلط (True / False)۔ سوال کے جواب سے پتہ چلتا ہے کہ آگے کس سمت میں جانا ہے۔ مثلاً

IS A>B? , IS X = 10?

فلوچارٹ کی بنیادی علامات اور ان کے افعال

علامات	نام	کام / استعمال
	آغاز / اختتام (Start / End)	بیضوی علامات آغاز یا اختتام کو ظاہر کرتی ہے۔
	تیر کا نشان (Arrows)	ایسی لائن علامات کو جوڑنے کے لیے استعمال ہوتی ہے اور علامات کے درمیان تعلق کو ظاہر کرتی ہے۔
	ان پُٹ / آؤٹ پُٹ (Input / Output)	متوازی الاضلاع کی علامت ان پُٹ یا آؤٹ پُٹ کو ظاہر کرتی ہے۔
	پروسیس (Process)	مستطیل کی علامت پروسیس کو ظاہر کرتی ہے۔
	فیصلہ (Decision)	ڈائمنڈ کی علامت فیصلے کو ظاہر کرتی ہے۔

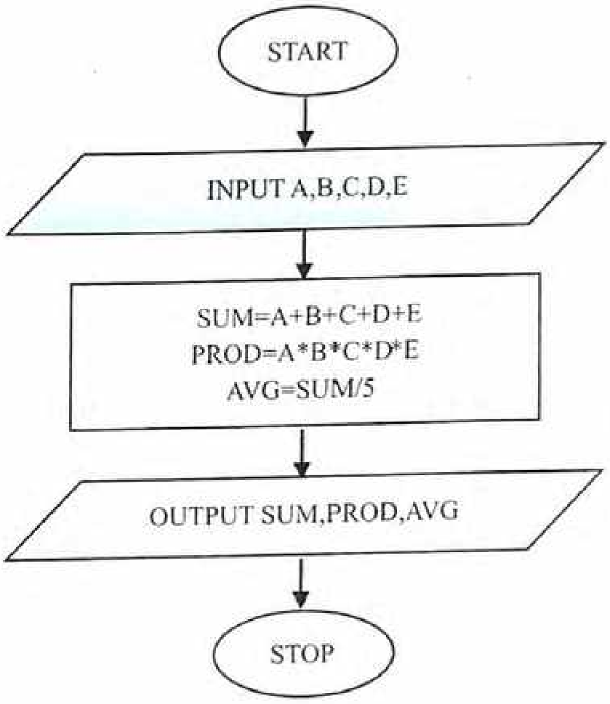
جدول نمبر 1.1 فلوچارٹ کی علامات

NOT FOR SALE

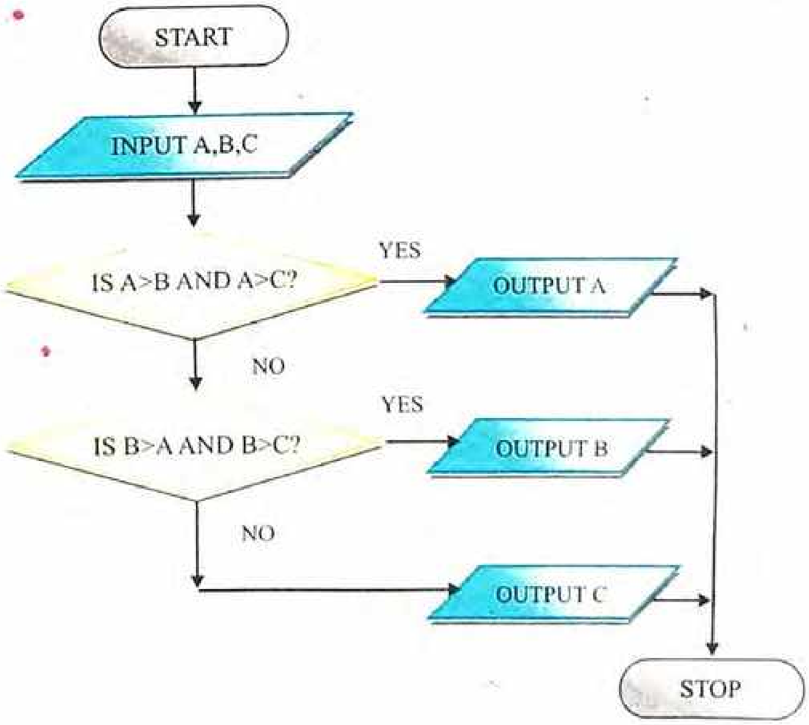
1.3.5 چند مسائل کے حل کے لیے فلو چارٹ (Flowcharts to Solve Some Problems)

اس حصے میں، پہلے حصے میں لکھے گئے چند الگورتھم کے لیے فلو چارٹ بنائے گئے ہیں تاکہ مسائل کے حل کے طریقہ کار کو باآسانی سمجھا جا سکے۔

فلو چارٹ نمبر 1: پانچ اعداد A,B,C,D,E کا حاصل جمع، حاصل ضرب اور اوسط معلوم کرنے کے لیے فلو چارٹ۔

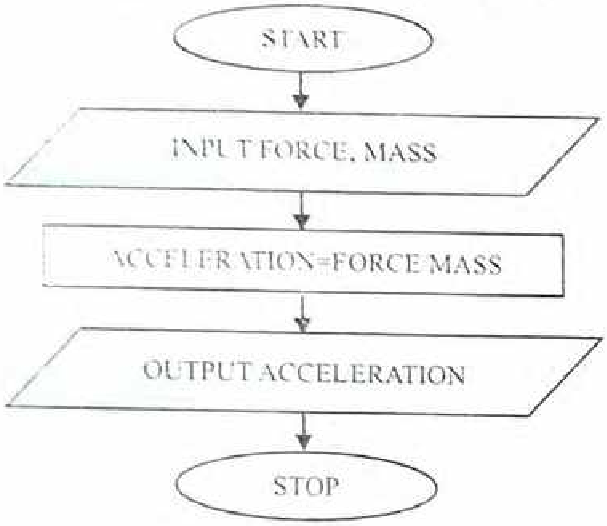


فلو چارٹ نمبر 2: تین غیر مساوی اعداد میں سے سب سے بڑا عدد معلوم کرنا۔

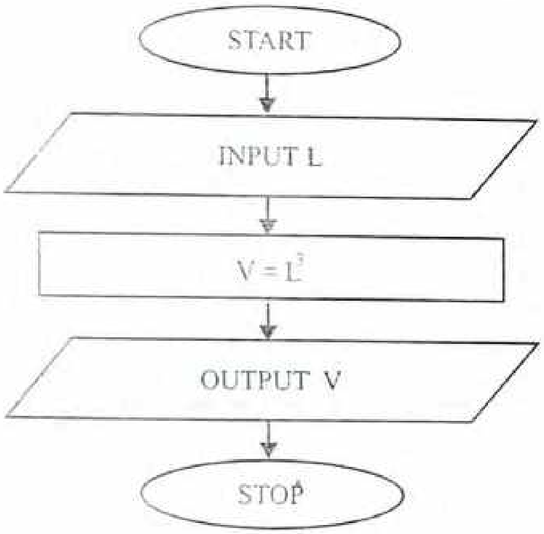


NOT FOR SALE

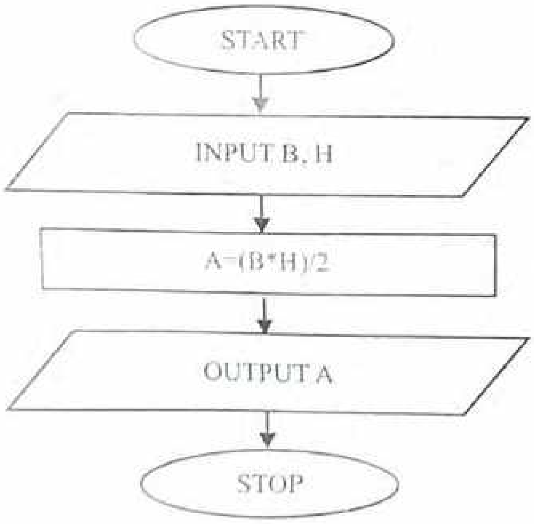
فلوچارٹ نمبر 3: کسی بھی متحرک جسم کا اسراع معلوم کرنا جبکہ اس کی کمیت اور عامل قوت معلوم ہو۔



فلوچارٹ نمبر 4: کسی بھی مکعب کے ایک ضلع کی لمبائی ان پٹ کر کے اس کا حجم معلوم کرنا۔

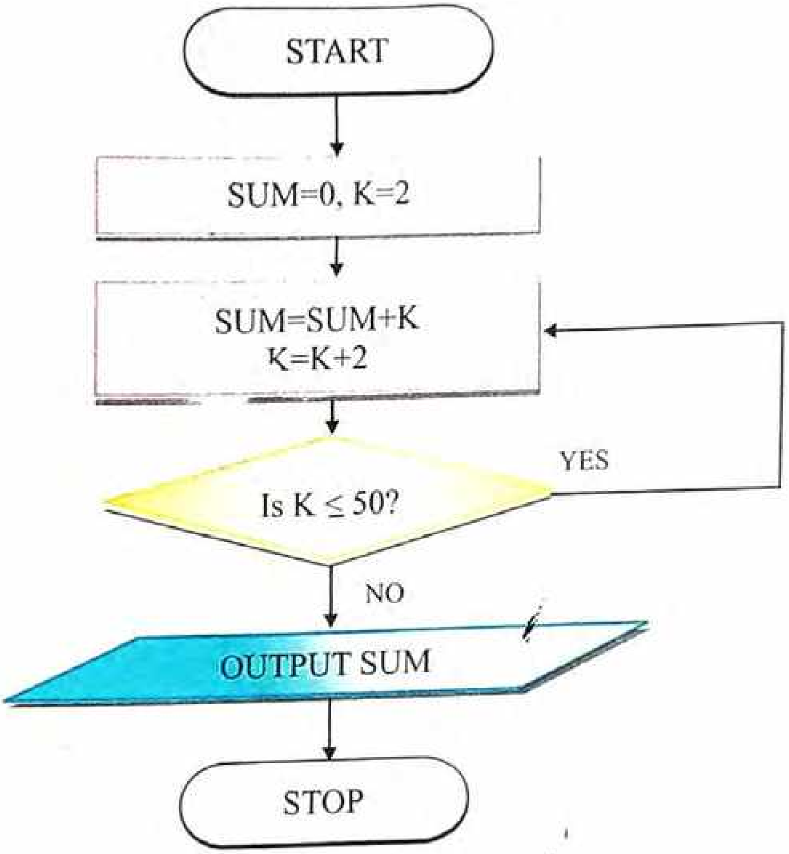


فلوچارٹ نمبر 5: کسی مثلث کا رقبہ معلوم کریں جبکہ اس کا قاعدہ اور اونچائی کی لمبائی دی گئی ہو۔

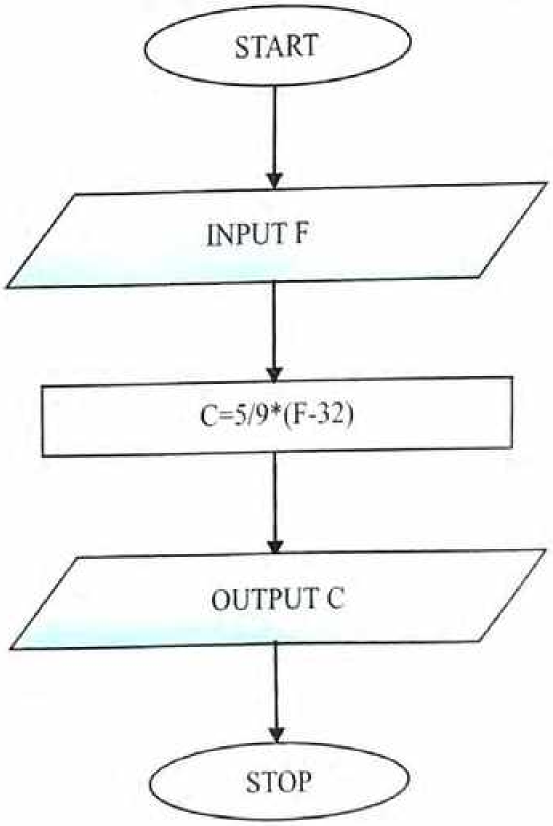


NOT FOR SALE

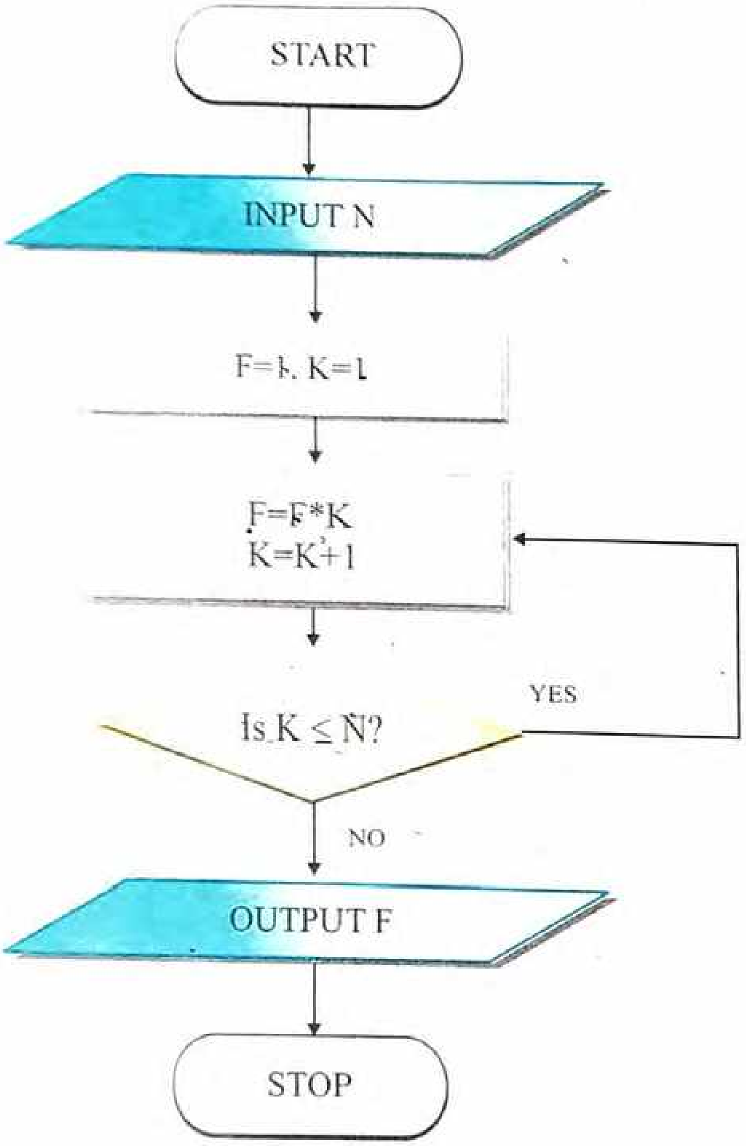
فلوچارٹ نمبر10: 1 سے لیکر 50 تک جفت اعداد کا مجموعہ معلوم کرنا۔



فلوچارٹ نمبر11: فارن ہائیٹ درجہ حرارت کو سینٹی گریڈ درجہ حرارت میں تبدیل کرنا۔



فلوچارٹ نمبر12: فیکٹوریل (Factorial) معلوم کرنا۔



اہم نکات

- مسئلہ ایک رکاوٹ، دشواری یا چیلنج یا ایسی صورتحال ہے جسے دور کرنے یا حل کرنے کی ضرورت ہو۔
- بہترین حل اسے کہا جائے گا جو کم لاگت، کم وقتی، تیز اور کم پیچیدہ ہو۔
- کسی بھی مسئلے کو حل کرنے کے کئی متبادل طریقے ہو سکتے ہیں۔
- مناسب اور منطقی اقدامات کی وہ فہرست جو کسی خاص مسئلے کو حل کرنے کے طور پر لکھی جاتی ہے اسے الگورتھم کہا جاتا ہے۔
- کمپیوٹر سائنس میں کسی بھی الگورتھم کی خصوصیات کو پرکھ کر ہم اس کی کارکردگی معلوم کر سکتے ہیں۔ الگورتھم کی خصوصیات میں ہم یہ جانچنے کی کوشش کرتے ہیں کہ کوئی بھی الگورتھم کس حد تک وسائل کو استعمال کرتا ہے؟

NOT FOR SALE

- مسئلے کو حل کرنے اور فیصلہ سازی کی صلاحیت الگورتھم کی استعداد کار کو بہتر بنانے اور بہترین نتیجہ حاصل کرنے کے لئے اہم ہنر ہے۔
- فلو چارٹ الگورتھم کو تصویری یا علامتی طور پر ظاہر کرتا ہے۔
- الگورتھم میں موجود مختلف اقسام کے اقدامات کے لئے فلو چارٹ کے اندر مختلف علامات استعمال ہوتی ہیں ان علامات کو فلو چارٹ کی علامات کہتے ہیں۔
- بیضوی / اوول شکل فلو چارٹ کے آغاز یا اختتام کو ظاہر کرتی ہے۔
- تیر کے نشانات ایک مرحلے سے دوسرے مرحلے تک کنٹرول کی تبدیلی کو ہر ظاہر کرتے ہیں۔ یہ ایک مرحلے سے دوسرے مرحلے تک پیش رفت کو بھی ظاہر کرتے ہیں۔
- متوازی الاضلاع، اِن پُٹ / آؤٹ پُٹ کو ظاہر کرنے کے لئے استعمال ہوتا ہے۔
- مستطیل پروسیسنگ کے مرحلے کو ظاہر کرتا ہے۔
- فلو چارٹ میں جب کوئی فیصلہ کرنا ہو یا کسی سوال کا جواب دینا ہو تو ایسی صورت حال میں ڈائمنڈ کی علامت استعمال ہوتی ہے۔ جیسا کہ ہاں یا نہیں یا صحیح یا غلط۔

مشق

صحیح جواب کا انتخاب کریں

__________ ایک رکاوٹ، مشکل یا چیلنج، یا کوئی بھی ایسی صورت حال ہے جس سے نکلنا مقصود ہو۔

الگورتھم   مسئلہ   (ج) پیچیدگی   (د) ان میں سے کوئی بھی نہیں

__________ الگورتھم کی ان خصوصیات کو بیان کرتی ہے جن سے ہم اس الگورتھم کے بارے میں یہ جاننے کی کوشش کرتے ہیں کہ الگورتھم کس حد تک وسائل کو استعمال کرتا ہے؟

الگورتھم کی کارکردگی   (ب) بہترین حل

الگورتھم کی پیچیدگی   (د) ان میں سے کوئی بھی نہیں

مناسب اور منطقی اقدامات کی وہ فہرست جو کسی مخصوص مسئلے کے حل کے طور پر لکھی جاتی ہے __________ کہلاتا ہے۔

الگورتھم   (ب) فلو چارٹ

الگورتھم کی پیچیدگی   (د) یہ تمام

NOT FOR SALE

.iv الگورتھم حسابی مرحلوں/اقدامات کا ایک ترتیب وار سلسلہ ہے جو ان پٹ کو ____________ میں تبدیل کرتا ہے۔

(الف) ڈیٹا سٹرکچر (ب) الگورتھم آؤٹ پٹ یہ تمام

.v ان میں سے کونسی علامت پروسیسنگ کو ظاہر کرتی ہے؟

(الف) (ب)

سوال نمبر2۔ درج ذیل کے مختصر جوابات تحریر کریں۔

.i "مسئلے کا تجزیہ" کی تعریف کریں۔

.ii مسئلے کے حل کی منصوبہ بندی کیسے کی جاتی ہے؟

.iii "مسئلے کا صریح حل" کی تعریف کریں۔

.iv مسئلے کے حل کے لیے کوئی سے تین طریقوں کی تعریف کریں۔

.v مسئلے کے بہترین حل کے چناؤ کے لیے کوئی سے تین عناصر کے نام لکھیں۔

سوال نمبر3۔ الگورتھم کی تعریف کریں اور مسئلے کے حل ...

سوال نمبر4۔ الگورتھم کی کارکردگی جانچنے کے لیے درج ذیل پیمانوں کی وضاحت کریں۔

.i درکار ان پٹ .ii پروسیسنگ کو مکمل کرنا

.iii فیصلہ کرنا .iv حاصل شدہ آؤٹ پٹ

سوال نمبر5۔ فلو چارٹ کیا ہے؟ فلو چارٹ کی علامات کو ...

سوال نمبر6۔ کسی بھی دیئے گئے عدد کا فیکٹوریل (Factorial) معلوم کرنے کے لیے الگورتھم لکھیں اور پھر اس کے لیے فلو چارٹ بھی بنائیں۔

سوال نمبر7۔ دو درجی مساوات (Quadratic Equations) ...

سوال نمبر8۔ کسی بھی طالب علم کے حاصل کردہ مضامین کا اوسط معلوم کرنے کے لیے الگورتھم لکھیں اور فلو چارٹ بنائیں۔

NOT FOR SALE