

question to be covered

1. C की विधियाँ

2. C की यूनिट शॉर्क्चरेस.

3. Token की समस्याएँ → In PDF

4. Header file को add करने के बारे मुद्रित हो।

5. Variable की Rules → In PDF

6. String वाला कोई 10-5 Library Function of

string.

7. Start Note → कंट्रोल, getch(), printf, scanf, Type casting,
short hand operator, pre/post decrement, Ternary
operator, fseek(), feil(), perror(), fprintf()

- 33 constant + types
- 34 Data types.
- 35 types of conversion.

8 Bitwise operator.

9. प्रिंटिंग नियम तथा एसीसीएचपी rule explain with syntax.

10. भूमिका तथा बिन्दुरी operators में अंतर
11. ऑपरेटर तथा कंस्ट्रुक्शन में अंतर
12. while एवं do while में अंतर
13. break एवं continue
14. प्रोग्राम का flow किसे पढ़ार से प्रारंभित प्रगति होती है।
15. what is control statement
16. nested Loop.
17. getch() तथा putch()
18. strcmp() stampi() में अंतर।
19. String की input करने के लिए किसे क्या कहते हैं?
20. पारंतर की विशेषता क्या है?
21. what is function
22. 'c' function में आरग्युमेंट ऑफन के प्रकार with exp
23. actual तथा formal आरग्युमेंट में अंतर
24. call by value or call by Reference को explain.
25. point to pointer to function का example दें।
26. storage classes.
27. global or local variable.
28. प्रिंटिंग सार साइरकेटिव वज्र होता है।
29. क्षेत्र तथा function में अंतर
30. goto statement
31. Array types.
32. c union.

Answers.

सीखने में वहुत आसान है।

- C Programming language
- C language में OS - के software कोड जैसे हैं OS, data base, computer drivers, Editors, Assembler, compiler : के लिए ही क्षमा गमा पर आय. अब C को unix OS सभी प्रकार के OS में काम आती है।
- Computer के भा अथ software Hardware
- कोड की ओरावर इन्सेप्ट होती है।
- 'C' general-purpose high-level-language है।

2. C का मूल structure.

```
#include <stdio.h> → Preprocessor
int main() → main function
{ → open curly brace
printf("Hello world"); → print i/o function.
return 0; → return 0;
} → closing curly brace.
```

1. preProcessor: program में सभी पहले preProcessors/header को प्रियो जाता है जो अलग-2 काम के लिए डिजिट कोड देते हैं।
For eg. stdio.h में printf और scanf function आते हैं।

2. int (main) → यहाँ पर main function का return type
 - integer → main function का return type void नहीं।
 - int का जट लिखा है।
 - void का जट लिखा है।
 - value का जट लिखा है।

जब कोड लिखता है Program की गुणात main() function की हीरिंग

3. ↳ इस function के code वाले statement की curly brace open होने के बाद लिखा जाता है।

4. printf → Statement की output में print किया जाता है।

5. return 0; यह प्रोग्राम को चल करने की अनुमति देता है।
नोट: main function की return करता है।

6. ↳ main function को close कर दिया है।

Question - 4

Header file को add करने के उपचार

- यहाँ Header file किसी विशेष operation को perform करता है।
- प्रोग्राम लिखने में कुछ समझ लगता है।
- compile time की speed increase
- code की readability को बढ़ा देता है।
- इसका extension .h होता है।
- इसमें #define standard library function के समूह को contain किया जाता है।
- #include का उपयोग Header file को program में include करने के लिए किया जाता है।

question 6)

- **String** → String character का एक ही है।
- यह one dimensional array होता है जिसमें
इनके character होते हैं।
- अखर) char Null ' \0' होता है।
- अब पुरा string इनमें होता है तो क्या " " में
इसके बारे में क्या होता है ?
- यह - यह character का इनमें होता है तो क्या " " में
इसका Data Type char होता है।

Syntex: char str1[100] = {"Hello"};

- में एक से अधिक character होते हैं इसलिए %s
String format specifier का उपयोग करते हैं। अब single
character print करना होता है %c

Function of String Lib.

1. **strcat()**: एक string की दूसरी string को जोड़ा जाता है।

strcat(destination_st, source_st);

2. **strcmp()**: दो string को compare किया जाता है, यह case-sensitive है।

strcmp(String1, String2);

3. **Strncmp()**: दो string को compare किया जाता है यह case-sensitive है।

4. strcpy() string की copy करना
strcpy(destination-string, source-string);
5. strdup() string की duplicate बुनौती के लिए!
• String - की string के जितकी duplicates अनावा भावांग!
6. strlen() string की length रखना!
strlen(string);
7. strlwr(); uppercase char की lowercase की convert
फिरा गिरा हो।
strlwr(string);
8. strrev(); string की reverse print करता हो।
strrev(string);
9. strupr(); Lowercase की character की uppercase की convert
करता हो।
strupr(string);

$$\begin{array}{r}
 1100 \\
 + 100 \\
 \hline
 100\text{ 0}1
 \end{array}$$

0 0 0 0

$$\begin{array}{r}
 1100 \\
 - 0100 \\
 \hline
 1010 \\
 - 101 \\
 \hline
 1010 \\
 - 101 \\
 \hline
 1010
 \end{array}$$

Bitwise operator:

Bitwise operator का उपयोग operands के बिंदु Bit level में ऑपरेशन perform करने के लिए आता है। जब ऑपरेटर को bit में convert करता है।

&	AND	<<	shift right-left
	OR	>>	shift right
^	XOR		
~	compliment		

Question q) precedence rule or associativity:

precedence rule से आवाय होने विभिन्न से होने वाले अभियंत्र के लिए इकाई के बीच पहले की ओपरेटर के अधिक ऑपरेटर के लिए पहले की ओपरेटर के अधिक ऑपरेटर का हल निकाला जाता है।

associativity:-

जब लकड़ियों के बीच की ओपरेटर के लिए अधिक ऑपरेटर होती है तो इसी सिद्धान्तिकी का नियम यह वर्ताता है कि पहले उसमें लिखी हए किमानों वाले से बाहरी वाले हो सकता है।

(1) [] ->

! ~ ++-- +- (Type)* & sizeof

* / %

+ -

<< >>

left to right

right to left

left to right

" " "

10

Unary operator

i) The operator which act upon single operand are called unary operator

ii) They are pre-increment, post increment

Binary operator

iii) The operator which requires two operands for their action are called binary operator.

iv) Arithmetic, relational, logical etc.

11

operator or expression

expression एक operand और operator की collection होती है। जो कि एक विशेष value की प्रस्तुत करता है।

e.g. $((45 + 5) / 2)$

operator वह है जो प्राप्त अवयव variable के द्वारा operation perform होता है।

12

while Loop

1. entry control loop है।
2. while loop की condition यह है कि बाहर की जानी है।
3. condition true होने की code execute होती है।

To while Loop

1. exit control loop है।
2. condition बाहर होने की जाती है।
3. condition true होने पर code execute होता है।

4. ; OR USE ~~FOR~~ ;

4. last ; OR USE,

Question 13) break and continue

```
#include < stdio.h>
int main()
{
    int i;
    for( i=1; i<=5; i++)
    {
        if( i == 3)
        {
            break;
        }
        printf("%d\n", i);
    }
    return 0;
}
```

while Loop Syntax :

variable initialization;

while (condition)

{

statement;

variable increment/ decrement

}

do {

statement;

}

while (condition);

do while Loop

Question (1) \Leftarrow Flow of program करने वाला नमूना - (3)

- Branching : if.
switch
conditional operator
- Looping : while
do while
for
- Jumping Statement : break
continue.

Question (1c) nested Loop \Rightarrow यह एक अलग लूप के अंदर अन्य लूप का उपयोग करते हैं तो उसे nested लूप कहते हैं।
जैसे लूप while, for, do while लूप एवं ऐसा है।

माना लूप अंदर की ओर के बाद लूप के बाहर
nested loop के अंदर ही loop के अंदर लूप
जैसा nesting कहते हैं।

for (initialize part ; condition part ; update part)
for (initialize part ; condition part ; update part)

// statement 1;
// statement 2;
:
// statement n;

Syntax (2) while loop

while (condition)

{

 while (condition)

{

 Statement;

{

 Statement;

}

 } body (block of code of nested loop)

getch or putch :

का फ़र्मेयल user का input मतलब 1 char होता है
getch की अनुप्राप्ति की जाती है।
यदि user ने कोई भी आवाज़ नहीं दी तो getch character input
तो putch पहले एक ही character का output में print
करता है।
का single char input function होता है जो conio.h file में

getch →
define होता है जो कि charter input लेता है तथा वह
screen पर echo नहीं होता है। मतलब किसी नहीं होता है।

Syntax →

getch();

or

#include <conio.h>
int getch(void);

putch : ~~int char at print~~

Syntax =

int putch(int c);

question 19) method 1. using gets.

syntax : `char * gets(char * str)`

OR

```
char str[20];
gets(str);
printf("%s", str);
```

method 2 using fgets:

ex, `fgets(str, 20, stdin)`

method 3) using `%[^n] %*c` inside scanf

ex, `scanf("%[^n] %*c", str);`

method 4. using `%[^n]s` inside scanf.

question 20 pointer पाइंटर का दोसा क्रीड़वाल है जो दुसरे वेरिएबल का एड्रेस स्टॉर करता है। अगर वेरिएबल किसी भी array, function TYPE का हो सकता है।

एक पाइंटर की size 4 बात होती है जिसे आपका compiler 64 bit वाला हो तो पाइंटर की size 8 byte होती है।

पाइंटर का data type उसके size पर असर नहीं डालता। पाइंटर जिस data type का होता है उसी लाइप के variable की & Address Point बनता है।

Syntax → `data-type * Var_name;`

```

return-type function-name(parameter1, parameter2 --- parameterN)
{
    // function body
    // ...
}
    "return value;"
```

→ syntax

about functions

एक statement का एक समूह होता है जो program के function होता है।

- function में लिखा code वार-वार नहीं लिखना पस्त।
- function programmer का सभी program की space
बचाता है।
- यह program को होर्टे function में विभागित किया जा सकता है।
- यह program में error जीवे में जैव आसानी से पाओल।
जा सकता है।
- उनके अनुसार function को वार-2 call किया जा सकता है।

Types-2: In predefined or in build function.

- इनको Library-function कहते हैं।
- इनके लिए #include header file या preprocessor जीवे होता है।
- Function की declaration, definition header file
i.e. printf, strlen.

2. user defined function.

- यह function जो user द्वारा जीवे उसकी आवश्यकता के अनुसार बनाते हैं।

function of call क्या है ?

परमेटर

Parameter

- Formal → वह function जिसकी declaration या definition है।
- Actual → वह function call जिसका उद्देश है।

1. call by value

- call by value method में actual parameter की value को function के formal parameter के copy करती है।
- इसमें बाहरी में तक ही इसमें variable की value का प्रभाग function call से प्राप्ति किया जाता है।
- call by value में ही formal parameter के लिए actual parameter की value नहीं बदल पड़ती।
- actual की formal parameter 3लाई-2 memory location में रखी रखती है।

2. call by Reference

- call by Reference में इस argument के address को formal parameter से copy किया जाता है।
- variable के address को actual para के address को pass करता है।
- formal parameter की value change एवं actual para की value change के लिए।
- actual parameter के memory location के लिए।

Pointer to function

जब हम किसी वरिएबल का Address पाहते हैं तो उसे भारतीय में directly
किसी function को pass करते हैं। तो यह Passing pointer to
function कहलाता है।

इस pointer की किसी function के argument के समान
में pass कर रहे हैं। Passing pointer to function,
passing pointer as argument, call by reference
इन सभी बातों के बारे में अच्छा है,

Syntex → return-type (*function-pointer-name)(parameter1,
parameter2 -- parameterN);

Storage class 1. C Automatic 3. C Register
2. C External 4. C Static

storage class variable का scope तथा lifetime set
करती है। कहाँ store करना तथा भारतीय में करती है।
storage class variable

1. C automatic.

→ auto keyword का use करते हैं इसमें

→ भी कई normal local variable होती है।

→ इसकी visibility function के अंदर होती है।

→ इनकी default value garbage होती है।

Syntex → auto data-type variable-name;

2. C External

- external storage class के 'extern' keyword का use होता है।
- variable का scope global होता है।
- इसकी default value '0' होती है।
- इसकी use program के external part तक function के अंदर नहीं होती।

Syntax - `extern data-type variable-name;`

3. C Register

- 'register' keyword
- Register storage class के variable का scope Local होता है।
- Register storage variable computer के register पर store होती है।
- register class का memory limited होती है। अग्रि
- Register की memory कम्प्युटर के बिट पर CPU पर store होती है।
- इनकी कोई address नहीं होती है।
- Default value garbage होती है।

Syntax → `register data-type variable-name;`

4. C Static

- static keyword का use करते हैं।
 - variable का scope Local अथवा global होता है।
 - default value '0'
- Syntax,

~~local~~ global

```
#include <stdio.h>
int g; → global variable
void main()
{
    // code
}
```

~~global~~ local

```
#include <stdio.h>
void main()
{
    int a; // local variable
    {
        // code
    }
}
```

Variable Scope - 2.

1 local variables, जहाँ function के अंदर होते हैं।

- उस function के अंदर होते ही पर visible रहते हैं।
- इनकी default value garbage value होती है।

2 global variables

→ global variable function के बाहर होती है।

→ global variable की visibility यह program में होती है।

→ default value o नहीं है।

conditional or ternary operators

→ इसमें तीन expression होते हैं।

→ अगर पहला expression true होता है तो वह दुसरा

expression output में print कर द्देता है।

→ अगर पहला expression false होता है तो वह तीसरा expression output में print करता है।

syntax → exp1 ? exp2 : exp3

preprocessor directive

जो C compiler को जाने से पहले

a program compiler

preprocess करता दिखा जाता है। जिसका काम preprocessor directive को process करना होता है। program को कोई भी बदलने जो # से शुरू होती है। preprocess directive कहलाती है।

#include
commonly use #include directive
to include header file in program at the
file में copy करता है।

#define -

program में constant या micro को define करने के लिए

#define PI 3.14;

#undef -

define constant या micro को undefine करने के लिए

#undefine PI;

#error

यदि program में macro को आवश्यक नहीं तो error macro
define करना या इस परिवर्तन से related
error show करने के लिए।

#ifdef square

#error please define the macro first

#endif

#pragma

- compiler को additional info देता है

macro - मेरे identifiers को ही program में वैध identifier
होता है वह पर कह Replace से उसकी जगह value की
लाती है।

predefined macro →

- DATE : date
- FILE : file name or path return
- LINE : first line पर के गढ़ी number
- TIME : current time & string return

macro

1. macro are preprocessed
2. no type checking is done

in macro

3. macro increase the code length

4. speed of execution using macro is faster

5. does not check any compile time error

6. macro useful when small code is repeated many times

7. Before compilation macro name is replaced by macro value

Function

1. functions are compiled.
2. done in function.

3. keep the code length unaffected.

4. slower the macro

5. checks compile time Error

6. at code point useful

7. during function call transfer of control takes place.

C Union.

- अलग-अलग data types का connection collection होते हैं
- union का उपयोग करना ही द्वारा union keyword का use करते हैं।
- Union में variable के decoration के member कहते हैं।
- भह प्रत्येक member के लिए अलग² memory allocate नहीं करता है।
- union के member के memory allocation share करते हैं।
- * Syntax union union-name {
 data-type member1;
 :
 :
 data-type member2;
 };

Type casting

- Type casting में एक variable के दूसरे variable के convert कर दिया जाता है।

[int] → [float]

Syntax

(data-type-name) expression.



Algorithm inc.

- अह निर्दिश का समूह है जिसके द्वारा computer Problem को solve किया जाता है।
- एल्गोरिदम का प्रयोग mathematics और computer science में समस्याओं की सुलबानी के लिए किया जाता है। यह जाकें कि इस के कारण में सक्षम होता है।
- Algorithm को Flowchart का जरूरी प्रयोग किया जाता है।
- Algorithm का प्रयोग हम किसी भी language के साथ।
- Algorithm के प्रयोग से Debugging का काम करता है।
- Algorithm

Example. Algorithm to find Area of triangle.

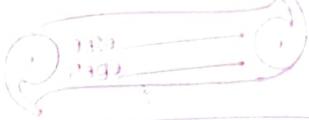
1. Start
2. Read base and Height of triangle
3. calculate the area of triangle.
4. Print area of triangle.
5. Stop.

Pseudo code

→ किसी कार्य को पूरा करने के लिए साधारण अंग्रेजी भाषा में कह कर में लिखे गए नियमों के समूह को pseudo code कहते हैं।

ex. of संख्याओं की विट का pseudo code.

1. write First number
2. write second number
3. store the sum of first number and second number
4. display sum.

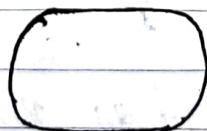


Flowchart

- किसी program या algorithm के क्रियों के बारा समझा जा पाएं वह कृदलाता है।
- इसके बारा कृदलाता है।
- इसमें वह सारे symbol का पुराग होता है भट्टा symbol
- program के flow की यदिक्षित करते हैं।
- step-by-step problem solving में help करता है किसी user किसी program के logic को जान पाए अथवा समझ पाए।

Used Symbols

1. **Start / Stop** → यह flow chart का पहला और अंतिम symbol है जिसका उपयोग किसी algorithm के शुरूआती तक अंत में किया जाता, जिसके नाम से भी जाना जाता है।
इसे terminal symbol के रूप में देखा जा सकता है।

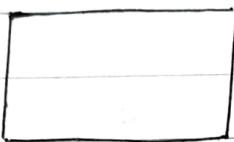


2. **Input / Output Symbol** → यह input / output या show के क्रमे के use में आता है।
इसके उपर्युक्त उदाहरण जैसा

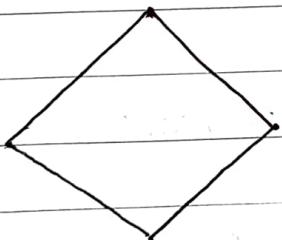


input - output बॉक्स

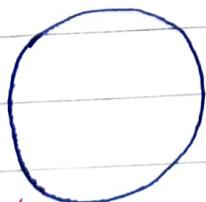
3. Processing symbol. - math की कामी +, -, ×, ÷ की
बताने हैं!



4. Decision symbol → True/False, yes/no जैसे प्रश्न
में के लिए यह दोहरा है।
→ if is shape like diamond



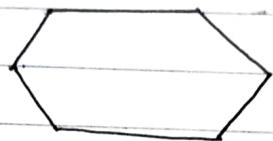
5. Connector flow chart के किसी भी उससे अलग भागों के जोड़ने के लिए।
→ if is round.



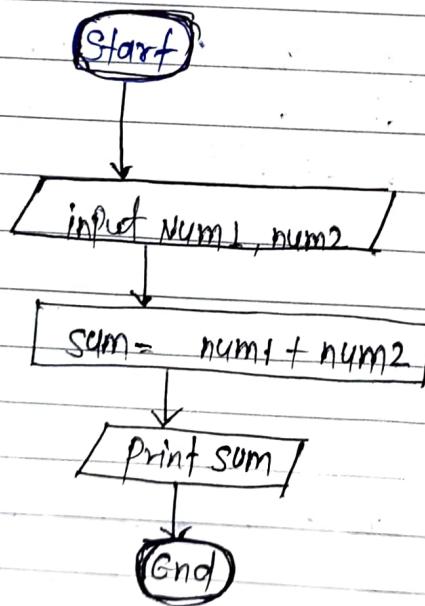
6. Flow line symbol → अप को विकित करने हेतु दिया जाता है,
down, left, right जैसा भी!

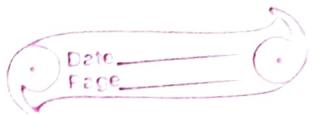


Hexagon symbols (उसी Preparation symbol) जो कहाँ आता है।
जैसे for loop, while loop; इन्हें Loop की योगिता कहते हैं।



Ex → Flow chart for adding two numbers.





C constant

- constant की value fix होती है।
(विरती) को ही सकता है।
- constant data type की जरूरत है।
- constant की lifespans भी कम हैं।
- constant pointer भी होता है।

constant एवं उसके लिये - 2)

1. Using `#define`:

```
#define CONSTANT_NAME constant_value
```

2. Keyword :-

```
const data_type constant_name = constant_value;  
                                                          * optional
```

Types :-

- integer 2, 10, -2 → normal value - 32768 → 32768
- decimal 2, 10, -2
- octal 02, 010
- Hexadecimal 0x12, 0x1f
- Floating Point -2.4, 4.8 → value " " ~~is~~
- character constant 'i', 'j'
- string constant Hel
- preprocessor #define a, s-

Data types

1. integer data type

- # variable कि प्र० प्र० घोषणा हैं int keyword use
- # numeric value store करती हैं
- # output print करती हैं %d format specifier
- # 2, 4, 8 bit की

// see in Prof. of 'C' Lecture

Type casting & Type conversion के बारे में

1. implicit →

इसका automatic type conversion कहते हैं।
भै जब होता है -

1. किसी data type अनुकूल हो।
2. उस द्वारे data type की value को data-type की लिए हो गी assign करते हैं।

Syntax new-data-type variable-name = value

2. Explicit type - manually type conversion कहते हैं। इसमें data-type को manually convert करता है।

- उस data की value को convert करते हैं।
- Data type compatible हो।



Syntax (target-type) expression
ex, double y = (double)n;

Debugging →

- Debugging का काम किया है जिसके द्वारा software code में से किया गया error और bugs को detect और correct किया जाता है।
- दूसरे शब्दों में यह कह सकते हैं कि जिसके द्वारा code में से error find करके उसे Analyze करके किया (fix) किया जाता है।

Top to down approach →

यह उपर से नीचे की ओर चलती है। इसमें पहले मुख्य समस्या डिफाइन की जाती है जिसे उस समस्या को ध्यान में रखकर आगे problem सॉल्व कि जाती है।

मुख्य problem को आगे sub - problem में विभाजित किया जाता है तथा उस sub - problem को solve करते हुए मुख्य problem को पुढ़वा दिया जाता है।

Symbolic constant → कानूनी होता है जो मैमोरी memory के location की represent करता है जहाँ पर value store होती है।

उसके नाम द्वारा आप value को access कर पाते हैं जैसे variable के समान करते हैं। उस value को किसी नहीं कर सकते हैं। symbolic constant को आप प्रोग्राम में कितनी भी बार use कर सकते हैं।

modular programming →

user

- इसकी अवधारणा 1960 के दशक में ~~programmer~~ की सहायता के लिए हुई थी।

- भै computer program को अल्गो-ड्राइवर्स में विभाजित करने की प्रक्रिया है।
- भै अप-प्रोग्राम आमतर पर function, module, processer, subroutines होते हैं।

main program

function 1

function 2

f3

f4

- पुरा प्रोग्राम एवं फ़ॉर्मूला में संग्रहित होता है।

- Program flexible बनता है। असाधारण स्पष्ट, कोड नियमों का संकेत है। modular programming