

(Q1) Distinguish between if-else statement and switch statement. Give example.

If else structure

1) If else statement uses multiple statement for multiple choices

2) If-else statement test for equality as well as for logical expression

3) If statement evaluate integer, character, pointer or floating point type or boolean type

4) Either if statement will be executed or else statement is executed

Switch statement

1) Switch statement uses single expression for multiple choices

2) Switch statement test only for equality

3) Switch statement evaluates only character of integer value

4) Switch statement execute one case after another till a break statement is appeared or the end of switch statement is reached

(Q2) Explain in brief structure of C program.

-> Structure is a user defined datatype in C language which allows us to combine data of different types together. Structure helps to construct a complex data type which is more meaningful. It is somewhat similar to an array, but an array holds data of similar type only. But structure on the other hand, can store data of any type, which is practice more useful. The components of the basic structure of a C program consists of 7 parts

- 1) Documents section: It is the section in which you can give comments to make the program more interactive the compiler won't complain & hence this portion would not be displayed on the output screen.
- 2) Preprocessor directives section: This section involves the use of header files that are to be included necessarily in the program.
- 3) Definition section: This section involves the definition and declaration in C.
- 4) Global declaration section: This section is used to define global variable to be used in the programs, that means you can use these variables throughout the programs that mean you can use these variables in all the functions.
- 5) Function prototype declaration section: This section gives the information about a function type, the parameters passed by the arguments.
- 6) Main function: It is the major section from where the execution of the program begins.
- Q3 State and explain while statement in C programming with suitable example.
→ While loop is a most basic loop in programming while loop has one control condition and execute as long as the condition is before the body of the loop or else hence it is called an entry-controlled loop. The while loop lets you repeat a statement until a specified expression becomes false.

include <stdio.h>

int main()

{

 int count = 1;

 while (count <= 10)

 {

 printf(" %d ", count);

 count++;

 }

}

i) What are the various datatypes available in C?
Explain the memory size and range of data possible.

→ The data-type in a programming language is the collection of data with having field as well as characteristics some of them are an integer, floating point, character etc. Usually programming language specify the range i.e. how gives data type.

C data types are used to:

- Identify the type of variable when it is declared.

- Identify the type of the return value of a function.

- Identify the type of a parameter expected by a function.

Neeraj Appar
F-129
28/3/13

A.L.N.R.A.

ANSI C provides 3 data types:
Primary (Built-in) Data types: void, int, char, double
and float

- 2) User-defined data types - Array, references and pointers
- 3) User-defined data types - Structure, union and enumeration

C data types	Storage size	Range
char	1	-127 to 127
int	2	-32,767 to 32,767
float	4	1E-37 to 1.E+37 (6)
double	8	1E-37 to 1.E+37 (10)
long double	10	1E-57 to 1E+37
long int	4	-2,147,483,647 to 2,147,483,647
short int	2	-32,767 to 32,767
unsigned int	2	0 to 65,535
short int	2	-2,147,483,647 to 2,147,483,647
Signed	2	-2,147,483,647 to 2,147,483,647
long int	4	0 to 4,294,967,295
unsigned long int	8	2 (power) 64 - 1

Q5 Differentiate between compilers and interpreters

Compilers

- 1) A compiler translates high level instruction directly into machine language.
- 2) The programs code is already translated into machine code. Thus, it is code execution.
- 3) You can't change the program without going back to the source code.
- 4) Compiler code runs faster.
- 5) It is based on language translation, linking, loading method.

Interpreters

- 1) An interpreter translates high level instruction into an intermediate form which it then executes.
- 2) Interpreters are easier to use especially for beginners.
- 3) Interpreted programs can run on computers that have the corresponding interpreter.
- 4) Interpreted code runs slower.
- 5) It is based on interpretation method.

Q6 Discuss the concept of nesting of loop with an example.

→ Nesting of loops is the feature in C that allows the looping of statement(s) inside another loop. Tests any number of loops can be defined inside another loop i.e., there is no restriction for defining any number of loops. The nesting level can be defined at n times so you can define any type of loop inside another loop.

```
Ex #include <stdio.h>
int main()
{
    int i, j;
    for (i = 1; i <= 10; i++)
    {
        for (j = 1; j <= 5; j++)
            printf ("%d\t", (i+j));
        printf ("\n");
    }
    return 0;
}
```

2) State and explain different datatypes available in C

- There are 3 datatype available in C
- Primary data types = void, int, char, double and float
 - void - It holds no values and is generally used for specifying the type of function
 - int - used to denote an integer type
 - char - used to denote a character type
 - float, double - used to denote floating type

3) Derived data type - Array, reference and pointer

- Arrays : Arrays are sequences of data having homogenous values. They have adjacent memory location to store values
- Reference : Function pointers allows refer-

function with a particular signature
pointers :- These are powerful C features which
are used to access the memory and
deal with their address

- 3) User defined datatype : Structure, union, Enum.
- Structure :- It is a package of variables of different types.
- Union :- These allow storing data types in the same memory location.
- Enum : Enumeration is a special data type that consists of integral constants, and each of them is assigned with specific name.

Q8 Explain with illustration working of Goto Statement

→ The Goto statement is rarely used because it's makes program confusing.

2) In C program goto statement jumps directly to the label mentioned in goto

#include<stdio.h>

```
int main()
{
    int sum=0;
    for (int i=0; i<=10; i++)
    {
        sum = sum + i;
    }
}
```

```
if (i==5)
    goto addition;
```

}

addition

```
print f("%d", sum);  
return 0;  
}
```

Output: 5

- Q. Write a short note on ternary operator
→ The ternary operator is an operator that exists in some programming languages, which takes three operands rather than the typical one or two. The most operator we. It's a shorter, a simple if else block. A ternary operator allows you to assign one value another if the condition is true and another if the condition is false. A ternary operator makes the assignment of a value to a variable easier to see because it's contained on a single line instead of an if else block.
Ex

Find Write a C program to find largest among two numbers using ternary operator

#include <stdio.h>

```
int main()
```

```
{
```

```
int n1=5, n2=10, max;
```

```
max (n1>n2)? n1 : n2;
```

```
printf ("largest number between %d and %d  
is %d", n1, n2, max);
```

```
return 0;
```

```
}
```

Nearaj Agarwal
E-120
3013

P-L-N-RAG

(Q10) What is null statement? explain a typical use of it.

A "null statement" is a statement containing only a semicolon; it can appear wherever a statement is expected or required. The correct way to code a null statement is, it is useful when the syntax of the language calls for a statement, but no expression evaluation. It consists of a semicolon. Null statements are commonly used as placeholders in iteration statements or as statements on which to place labels of the end of compound statements or functions.

CH/021 Programming with C
Unit 2

- ai How function declaration differs from function definition? write a C program to show calling by passing by value.
In other words, a function declaration declares the name of the function and its type of what it returns you must declare identifier before you can use it. A function prototype is a declaration of a function that declares this type of the function parameters. A function definition defines the function itself.
2) C program to show to swap the values of two variable using call by value.

```
#include <stdio.h>
void swap (int,int);
int main()
{
    int a=10;
    int b=20;
    printf ("Before swapping a=%d, b=%d\n",a,b);
    printf ("After swapping a=%d, b=%d\n",a,b);
    swap (a,b);
}
void swap (int a, int b)
{
    int temp;
    temp = a;
    a = b;
    b = temp;
```

Point 8 ("After sweeping" $a = [0, 1, 6, 0, 1]$, $b = [0, 1, 1, 0, 1]$)

Output:

Before swapping $a = [1, 6, 2, 0]$

After swapping $a = [2, 0, 1, 6]$ in function $a = [2, 0, 1, 6]$

After swapping $a = [1, 0, 6, 2]$

- Q 2 What are multidimensional arrays? How can we access an array element?
Explain with the help of an example.
- If a multidimensional array is an array with more than two dimensions. In a matrix, the two dimensions are represented by rows & columns.

→ We know that array elements are randomly accessed using the subscript value. Array can be accessed using array name and subscript written from inside pair of square brackets.
example

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void main()
```

```
{
```

```
int arr[5] = {51, 52, 43, 25, 5, 26};
```

```
int i;
```

```
for (i=0; i<5; i++)
```

```
{
```

```
    printf("Element at arr[%d] is %d", i, arr[i]);
```

```
}
```

Notes of APPL
FYCG
O 4/10/20

B. L. A. Rao

Output:

Element at arr[0] is: 51 : Element at arr[3] is:

Element at arr[1] is: 32 : Element at arr[4] is 5

Element at arr[2] is 43 : Element at arr[5] is 8

a3 Briefly explain the working of following functions : a) getch(), fput()

; a) getch() : getch is a function in C programming language that reads a single character from the standard input stream. It is used to read what it is, and return it to the program. It is specified in ANSI-C and is the most basic input function in C. It is included in the stdio.h header file. The following program uses getch() to read characters into an array and print them out using the putchar function after one end of each character is found.

#include <stdio.h>

int main (void)

{ char str [100];

int ch, n=0;

while ((ch=getchar())!=EOF && n<100)

{ Str [n]=ch;

t=n;

}

for (n=t; i=0; i++)

putchar (str [i]);

Q3. parts(): The parts() function is very much similar to printf() function. The parts() function is used to print the string on the console which is previously read by using gets() or scanf(). The parts() function takes integer value representing the number of characters being printed on the console. Since it prints an additional newline character with the string, which moves the cursor to the newline console.

Q4. Write a short note on global and local variable.

- 1) Global variables as the name implies, are variables that are accessible globally or everywhere throughout the program. Once declared they remain in memory throughout the runtime of the program. This means that they can be changed by any function.
- 2) Local variable - The local variable declaration explicitly defines the type of the variable that has been declared along with the identification that names the variable. A local variable is a type of variable that can be used where the scope is extended.

Q5 What is an array? Advantages and disadvantages of an array.

→ In computer science an array is a data structure consisting of a collection of elements, each identified by a unique index or key.

i) Advantages of array

- Collection of similar types of data.
- We have only rememberable characteristics of array.
- Used to implement other data structures like linked list, stacks etc.
- 2 Dimensional array is used to represent a matrix.

ii) Disadvantages of array :-

- Time complexity increases in insertion and deletion operation.
- Wastage of memory because arrays are fixed in size.
- If there is enough space present in the memory but not in contiguous form, in this case you will not able to initialize your array.
- It is not possible to increase the size of the array once you had declared the array.

Q6 Write a C program to find largest and smallest numbers in an array.

→ #include <stdio.h>

```
int main ()  
{
```

```
    int arr[10], size, i, minimum, min-position, maximum, max-position;  
    printf ("In please enter the size of array: ")  
    scanf ("%d", &size);
```

Neeraj Appari
F-129
CH/OU

A.L.N.RAO

print("Please enter %d elements of an array: ",
size);

for (i=0; i<size; i++)

scanf("%d", &a[i]);

{
minimum = a[0];

maximum = a[0];

for (i=1; i<size; i++)

{
if (minimum > a[i])

{
minimum = a[i];

min-position = i;

if (maximum < a[i])

{
maximum = a[i];

max-position = i;

{
}

print("Smallest element in an array = %d, minimum = %d",
minimum);

print("Largest element in an array = %d, maximum = %d",
maximum);

return 0;

{

Output

Please enter size of an array: 4

Please enter 4 elements of an array:

10 20 30 40

Neeraj Agarwal
9-12th
04/04

Page No.
Date

P.L.N.R.D

smallest element in array is

largest element in an array is

Write a C program using recursive function to find Fibonacci series up to a number accepted from user

? #include <stdio.h>

int f(int);

int main()

{

int n, i=0, c;

scanf("%d", &n);

printf("fibonacci series terms are = %d", n);

for (c=1, cc=n; c<r)

printf("%d\n", f(i));

i++;

}

return 0;

}

int f(int n)

{ if (n==0 || n==1)

return n;

else

return (f(n-1) + f(n-2));

}

Never appear
for 12th
04/04

A.L.N.RAO

(Q) Define recursion, write a program to calculate factorial of a number entered by user using recursion.

→ Recursion is a program technique that allows the programmer to express operation in terms of itself. In C, this takes the form of a function that calls itself.

#include <stdio.h>

```
long int multiplynumbers (int n);
main()
int n;
printf ("Enter a positive integer: ");
scanf ("%d", &n);
multiplynumbers (n);
return 0;
}
```

long int multiplynumbers (int n)

{ if (n == 1)

return n * multiplynumbers (n - 1);

else

return 1;

}

Output

Enter a positive integer: 6
Factorial of 6 is 720

a) Explain the following declaration syntax and examples.

1) pointer to array = pointer to an array is also known as array pointer. By using the pointer to access the contents of the array.

Syntax = Data-type (* or > n) (size of array)
Ex - int (* p)();

2) Array to pointer = When an array is declared compiler allocates sufficient amount of memory to contain all the elements of the array. We can also declare a pointer of type int to point to the array like int * p; or int p[];
Ex - int p[3]; Both the statements are equivalent.

b) Explain the different categories of function.
Write a program to explain any one

1) There are 4 categories of functions -
1) Function with no argument and no return value

2) Function with no argument but return a value

3) Function with argument but no return value

4) Function with argument and return a value

1) Function with no argument and no return value
To create a function with no argument and no return value we set the parameter list as void and return type of function to void()

2) Function with no argument but return a value:
For this type of function we have void in the parameter list, but we set the return type to match the type of value returned by the function.

3) Function with argument but no return value:
In this type of function we have the return type set to void.

4) Function with argument and return a value:
In this type of function we'll both the parameter list and the return type.

Q) C program for function with no argument or no return value

#include <stdio.h>

void sum(void):

void main()

{

 sum();

}

void sum(void)

{

 float year = 1, interest = 5, amount = 5000, sum = 0;

 float sum;

 while (year <= 10) {

 sum = sum * (1 + interest);

 year = year + 1;

}

Nearby Appar.

Página

A.L.N.R.A.C

PrinR C¹⁴T4 total amount is 0.021 (sim),

Output

The total amount is \$600,000,000

Programming with C (Contd)

(Q1) How structures are initialized? How do structures differ from array?

→ C language supports multiple ways to initialize a structure, you can use any of the initialization method to initialize your structure.

- Initialize using dot operator
- Value initialized structure variable
- Variant of value initialized structure variable

Array

- 1) An array is a collection of variables of same data type stored in contiguous memory location.
- 2) Array elements are accessed by their index number.
- 3) Every element in array is of same size.
- 4) There is no keyword to declare an array.

Structure

- 1) A structure is a collection of variables of different data types.
- 2) Structure elements may not be stored in contiguous memory location.
- 3) Structure element are accessed by their name.
- 4) Every element in a structure is of different data type.

- 5) Struct is a keyword of C to declare the structure.

(Q2) What are pointers? State and explain benefits of using pointers.

- 1) Pointer in C language is a variable that stores the address of another variable.
- 2) A pointer in C is used to allocate memory dynamically i.e. at run time.
- 3) It allows management of structure which are allocated memory dynamically.
- 4) It allows passing of array and strings to function more efficiently.
- 5) It makes possible to pass address of structure instead of entire structure to the function.
- 6) It makes possible to return more than one value from the function.

(Q3) What are nested structures? State and explain with an example.

Nested structure in C is nothing but structure within structure. One structure can be declared inside other structure as we declare structure variable, instead of a structure.

#include <stdio.h>
struct address

```
    {char city[20];  
     int pin;  
     char phone[14];}
```

Struct employee

{
char name [20];

Struct address add;

} ;

Void main()

{

```
    struct struct employee emp;
    printf ("Enter employee information \n");
    Scanf ("%S%[S]S", &emp.name, &emp.add.city,
           &emp.add.pin, &emp.add.phone);
    printf ("Printing the employee information \n");
    printf ("name %s in city : %s pincode : %d phone
           %s", &emp.name, &emp.add.city, &emp.add.pin,
           &emp.add.phone);
```

}

Output: Enter employee information!

Yash

Mumbai

110023

2582562590

printing the employee information ..

name : Yash

City : Mumbai

pincode : 110023

phone : 2582562590

Q1) Explain malloc() and calloc() with syntax & example. How are they different?

→ 1) malloc(): allocates requested size of bytes and return void pointer pointing to the first byte (blk) after allocated space.

2) Syntax - void* malloc (byte size)

Ex - int *x;

x = (int*) malloc (50 * sizeof(int));

True (1);

2) calloc(): allocates space for an array of elements. Initialize them to zero and return a void pointer to the memory.

Syntax - void* calloc (number of items, element size)

Ex - Struct employee

{ char t name;

int salary;

type all Struct employee emp;

emp * e1;

e1 = (emp*)calloc (30, sizeof (emp));

Q3) What is use of dynamic memory allocation?

→ Dynamic memory allocation is an aspect of allocating and freeing memory to you. It is used in leads. Dynamic memory is managed and served with pointers that point to the newly allocated space of memory in an array which we call the heap. Now you can create and destroy an array of elements at anytime.

without any problems. To sum up, the automatic memory management uses the stack, and the dynamic memory management uses the heap. The (talloc.h) library has a function responsible for dynamic memory management.

(Q6) Write a note on unions in C support your answer with an example

A union is a special data type available in C that allows to store different data types in the same memory location. You can define a union with many members, but only one member can be used at a given time. Union provide an efficient way of using the same memory location for multiple purposes.

#include <stdio.h>

#include <string.h>
union Data {

int i;

float f;

char str[20];

}

int main () {

union Data type;

printf ("Memory size occupied by data : %d\n",
size of (data));

return 0;

3

Output

Memory size occupied by data: 20

(07) What is file? list out and explain the two different functions available for performing file related operations.

→ A file represents a sequence of bytes on the disk, where a group of related files is stored. File is created for permanent storage of data. If it is ready made structure. The most basic operations that program can perform on a file are:

- 1) Create a new file.
- 2) Opening an existing file (fopen)
- 3) Reading from file (fscanf or fgets)
- 4) Writing to a file (fprintf or fputs)
- 5) Moving to a specific location in a file (seek, rewind)
- 6) Closing a file (fclose)

(08) Define fopen() function in detail.

→ fopen() function is used to open a file & performs operation such as reading/writing & In a C program we declare a file pointer. Use fopen() as below. fopen() function creates a new file if the mentioned file name does not exist.

file fp;
fp = fopen("filename", "mod");

Rp - file pointer to the data type 'FILE' filenam
file actual file name with full path of file
mode - refers to the operation that will
be performed on the file. Example: `W`rite,
`R`ead and `C`oncatenate. Below is the
description for those mode of operation.

- Q9) Explain any 2 library functions provided in C language to manipulate strings.
- 1) `strcat()`: The function `strcat()` concatenates two strings.
 - The `strcat()` function is defined in `<String.h>` header file.
 - `strcat()` Prototype:
`char *strcat (char *dest, const char *src);`
It takes two arguments, i.e. two strings or character array, and stores the resultant concatenated string in the first string specified in argument.
 - 2) `strlen()`: The `strlen()` function calculates the length of a given string.
 - The `strlen()` function takes a string as an argument and returns its length. The returned value is of type long int.
 - It is defined in the `String.h` header file.

- Q10) Write a C program to copy the content of one file into another.
- ```
#include <stdio.h>
```

```
FILE *Rptr1, *Rptr2;
char filename[100];
printf ("Enter the file name to open for reading");
scanf ("%s", filename);
Rptr1 = fopen (filename, "r");
if (Rptr1 == NULL)
{
 printf ("cannot open file %s\n", filename);
 exit (0);
}
```

```
printf ("Enter the filename to open for writing");
printf ("Enter the filename to open for writing");
scanf ("%s", filename);
Rptr2 = fopen (filename, "w");
if (Rptr2 == NULL)
{
 printf ("cannot open file %s\n", filename);
 exit (0);
}
```

```
C = fgetc (Rptr1);
while (C != EOF)
{
 putc (C, Rptr2);
 C = fgetc (Rptr1);
}
```

```
printf ("Content copied to %s", filename);
```