

Model Answer for

B.Sc, computer Science, Integrated U.G/P.G(3rd Semester) Examination,2013

Subject: Introduction C language

(PCSC-302)

[Time: Three Hours]

[Maximum Marks : 30]

Question 1 (4 x 1 marks) and 2 (3 x 2 marks) are compulsory. Attempt any 4 among 3 to 9 (4 x 5 marks).

---

1. State true or false.

- 1.1. Each new c instruction has to be written on a separate line. **[FALSE]**
- 1.2. Capital letters can be used in variable name. **[TRUE]**
- 1.3. % operate cannot be used on float.**[TRUE]**
- 1.4. The maximum value that an integer constant can have varies from one compiler to another.**[TRUE]**

2. Find the correct output.

2.1. #include<stdio.h>

```
void main()  
{  
printf ( "nn \n\n nn\n" );  
printf ( "nn /n/n nn/n" );  
}
```

OUTPUT:-

```
nn  
  
nn  
nn /n/n nn/n
```

2.2. #include<stdio.h>

```
void main()  
{  
int i = 4, j = -1, k = 0, w, x, y, z ;  
w = i || j || k ;  
x = i && j && k ;  
y = i || j && k ;  
z = i && j || k ;  
printf ( "\nw = %d x = %d y = %d z = %d", w, x, y, z ) ;  
}
```

OUTPUT:-

```
w=1 x=0 y=1 z=1
```

```

2.3. #include<stdio.h>
void main()
{
int x = 4, y, z ;
y = -x ;
z = x-- ;
printf ( "n%d %d %d", x, y, z ) ;
}

```

OUTPUT:-

```
n233
```

3. Write a program to check whether the given number is a prime.

Ans: Several programs are possible depending on logic. A sample program is following.

```

#include<stdio.h>
void main()
{
int n,i;
printf("Enter number:");
scanf("%d",&n);
for(i=2;i<n/2;i++)
{
if(n%i==0)
{
printf("Number is not prime");
break;
}
}
if(i==n/2)
printf("number is prime");
}

```

4. Write a program to add two matrices of dimension (3X3).

Ans: Following is a program that serves the above purpose.

```

#include<stdio.h>
void main()
{
int a[3][3],b[3][3],c[3][3],i,j;
printf("Enter the First matrix->");
for(i=0;i<3;i++)
for(j=0;j<3;j++)
scanf("%d",&a[i][j]);
printf("\nEnter the Second matrix->");
for(i=0;i<3;i++)
for(j=0;j<3;j++)
scanf("%d",&b[i][j]);
}

```

```

printf("\n\nThe First matrix is\n\n");
for(i=0;i<3;i++)
{
    printf("\n\n");
    for(j=0;j<3;j++)
        printf("%d\t",a[i][j]);
}
printf("\n\nThe Second matrix is\n\n");
for(i=0;i<3;i++)
{
    printf("\n\n");
    for(j=0;j<3;j++)
        printf("%d\t",b[i][j]);
}
for(i=0;i<3;i++)
    for(j=0;j<3;j++)
        c[i][j]=a[i][j]+b[i][j];
printf("\n\nThe Addition of two matrix is\n\n");
for(i=0;i<3;i++){
    printf("\n\n");
    for(j=0;j<3;j++)
        printf("%d\t",c[i][j]);
}
}
}

```

5. Write a program to find maximum and minimum element in an one dimension array.

Ans: Following is a program which will serve nicely for above purpose. Other logics are acceptable.

```

#include<stdio.h>
void main()
{
    int a[20],i,max,min; // declared the array a with size 20
    clrscr();

    printf("\n Enter 20 numbers of elements for 1-D array : ");
    for(i=0;i<n;i++)
    {
        printf("\n Enter element [%d] : ",i+1);
        scanf("%d",&a[i]);
    }
    max=min=a[0];
    for(i=0;i<20;i++)
    {
        if(max<a[i])
            max=a[i];
        if(min>a[i])
            min=a[i];
    }
    printf("\n Greatest element from above array inserted is : %d",max);
    printf("\n Smallest element from above array inserted is : %d",min);
}

```

```
    getch();  
}
```

6.

6.1. State any four uses of C language.[2]

Ans: Following are some of the uses of C language

- Major parts of popular operating systems like Windows, UNIX, Linux is still written in C. This is because even today when it comes to performance (speed of execution) nothing beats C. Moreover, if one is to extend the operating system to work with new devices one needs to write device driver programs. These programs are exclusively written in C.
- Mobile devices like cellular phones and palmtops are becoming increasingly popular. Also, common consumer devices like microwave oven, washing machines and digital cameras are getting smarter by the day. This smartness comes from a microprocessor, an operating system and a program embedded in this devices. These programs not only have to run fast but also have to work in limited amount of memory. No wonder that such programs are written in C. With these constraints on time and space, C is the language of choice while building such operating systems and programs.
- You must have seen several professional 3D computer games where the user navigates some object, like say a spaceship and fires bullets at the invaders. The essence of all such games is speed. Needless to say, such games won't become popular if they takes a long time to move the spaceship or to fire a bullet. To match the expectations of the player the game has to react fast to the user inputs. This is where C language scores over other languages. Many popular gaming frameworks have been built using C language.
- At times one is required to very closely interact with the hardware devices. Since C provides several language elements that make this interaction feasible without compromising the performance it is the preferred choice of the programmer.

6.2. What is the structure of C program.[3]

Ans: Student should illustrate the structure with help of a sample program. Following is a sample program. It would simply print "Hello World".

```
/* comments*/  
/* Program to print Hello World */  
#include<stdio.h>  
main( )  
{  
printf("Hello World");  
}
```

Students should explain each part of the program with emphasis. For example we are explaining main function.

**main( )** is a collective name given to a set of statements. This name has to be **main( )**, it cannot be anything else. All statements that belong to **main( )** are enclosed within a pair of braces { } as shown below. main( )

```
{  
statement 1 ;  
statement 2 ;  
  
statement 3 ;
```

}

- Technically speaking **main( )** is a function. Every function has a pair of parentheses ( ) associated with it. We would discuss functions and their working in great detail in Chapter 5.

- Any variable used in the program must be declared before using it. For example,

```
int p, n ;  
float r, si ;
```

- Any C statement always ends with a ;

For example,

```
float r, si ;  
r = 8.5 ;
```

- In the statement,

```
si = p * n * r / 100 ;
```

\* and / are the arithmetic operators. The arithmetic operators available in C are +, -, \* and /. C is very rich in operators. There are about 45 operators available in C. Surprisingly there is no operator for exponentiation... a slip, which can be forgiven considering the fact that C has been developed by an individual, not by a committee.

- Once the value of **si** is calculated it needs to be displayed on the screen. Unlike other languages, C does not contain any instruction to display output on the screen. All output to screen is achieved using readymade library functions. One such function is **printf( )**. We have used it to display on the screen the value contained in **si**. The general form of **printf( )** function is,

```
printf ( "<format string>", <list of variables> ) ;
```

<format string> can contain,

%f for printing real values

%d for printing integer values

%c for printing character values

In addition to format specifiers like **%f**, **%d** and **%c** the format string may also contain any other characters. These characters are printed as they are when the **printf( )** is executed.

Following are some examples of usage of **printf( )** function:

```
printf ( "%f", si ) ;  
printf ( "%d %d %f %f", p, n, r, si ) ;  
printf ( "Simple interest = Rs. %f", si ) ;  
printf ( "Prin = %d \nRate = %f", p, r ) ;
```

The output of the last statement would look like this...

```
Prin = 1000  
Rate = 8.5
```

What is '\n' doing in this statement? It is called newline and it takes the cursor to the next line.

Therefore, you get the output split over two lines. '\n' is one of the several Escape Sequences available in C. These are discussed in detail in Chapter 11. Right now, all that we can say is '\n' comes in handy when we want to format the output properly on separate lines.

**printf( )** can not only print values of variables, it can also print the result of an expression. An expression is nothing but a valid combination of constants, variables and operators. Thus, 3, 3 + 2, c and a + b \* c - d all are valid expressions. The results of these expressions can be printed as shown below:

```
printf ( "%d %d %d %d", 3, 3 + 2, c, a + b * c - d ) ;
```

Note that **3** and **c** also represent valid expressions.

Students should explain other parts similarly.

7.

7.1. Explain if-else statement with syntax.

Ans: The C language provides a way to implement decision making in programs. Using If-else we can execute a block of statements according to the condition they satisfy.

Syntax:

```
if(condition)
{
statements to be executed in case the condition is true.
}
else
{
Statements to be executed in case the condition is false.
}
```

Students should explain the meaning of words in syntax.

7.2. Write a program to find the greatest of three numbers entered through keyboard.

Ans: Following is a sample program to serve the above purpose. Other logics are acceptable.

```
#include<stdio.h>
void main()
{
    int a,b,c;
    printf("please enter three numbers separated with space ");
    scanf("%d%d%d", &a,&b,&c);
    if(a>b)
    {
        if(a>c)
        {
            printf("a is greatest among three");
        }
        else
        {
            if(a==c)
            {
                printf("a and c are equal and greater than b");
            }
            else
            {
                printf("c is greatest among three");
            }
        }
    }
}
else
{
    if(b>c)
    {
        printf("b is greatest among three");
    }
    else
    {
        if(a==b)
        {
```

```

        if(b==c)
        {
            printf("a b and c all are equal");
        }
        else
        {
            if(c>b)
            printf(" C is greatest among the three");
        }
    }
    if(b==c)
    {
        printf("b and c are equal and greater than a");
    }
    else
    {
        printf("c is greatest among three");
    }
}
}
}

```

8.

8.1. Explain Do-While loop statement with syntax.

Ans: Do- while loop statement is a statement which is used to iterate a block of statements till a specified condition is fulfilled. Do-while loop provides a facility of executing the block of statement at least once even if condition cannot be satisfied.

Syntax:

```

do
{
Statement 1;
statement 2;
.
.
Statement n;
}while(condition);

```

Students should explain each written word in the syntax.

8.2. What is difference between break and continue.

Ans:

Break	Continue
When encountered, control automatically passes to the first statement after the loop.	When encountered, control automatically passes to the beginning of the loop.
Used with Switch case and loops.	Used with loops only.

Syntax	Syntax
An example	An example.

Students should have written at least these difference and explain wherever needed.

9.

9.1. How to initialize two dimensional array.

Ans: Following is the syntax to initialize an array.

<data type> <array name> [row size][col size]={values in 1<sup>st</sup> dimension}, {values in 2<sup>nd</sup> dimension}}

example:

```
int marks[2][3]={{20,30,10},{12,23,17}};
```

9.2. Find the transpose of a matrix.

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

```
void main()
```

```
{
```

```
int m, n, c, d, matrix[3][3], transpose[3][3];
```

```
printf("Enter the elements of matrix \n");
```

```
for( c = 0 ; c < 3 ; c++ )
```

```
{
```

```
for( d = 0 ; d < 3 ; d++ )
```

```
{
```

```
scanf("%d",&matrix[c][d]);
```

```
}
```

```
}
```

```
for( c = 0 ; c < 3 ; c++ )
```

```
{
```

```
for( d = 0 ; d < 3 ; d++ )
```

```
{
```

```
transpose[d][c] = matrix[c][d];
```

```
}
```

```
}
```

```
printf("Transpose of entered matrix :-\n");
```

```
for( c = 0 ; c < 3 ; c++ )
```

```
{
```

```
for( d = 0 ; d < 3 ; d++ )
```

```
{
```

```
printf("%d\t",transpose[c][d]);
```

```
}
```

```
printf("\n");
```

```
}
```

```
}
```