

12/12/19

96

This question paper contains 8+2 printed pages]

Roll No.

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

S. No. of Question Paper : 7507

Unique Paper Code : 32371502 J

Name of the Paper : Statistical Computing Using C/C++  
Programming

Name of the Course : B.Sc. (H) Statistics

Semester : V

Duration : 3 Hours

Maximum Marks : 75

(Write your Roll No. on the top immediately on receipt of this question paper.)

All questions are compulsory.

1. Attempt any ten parts :  $10 \times 3 = 30$

(i) State whether the following statements are true or false :

- (a) The expression  $*p++$  increments p.
- (b) Pointer stores only an integer value.
- (c) The left operand in an assignment expression must be a single variable.
- (d) Any expression that evaluates to an integral value may be used as an index.

P.T.O.

- (e) Dynamically allocated memory can only be referred to through pointers.
- (f) Adding one to pointer increases the address stored in it by one byte.
- (ii) Fill in the blanks :
- (a) If a function calls itself it is called \_\_\_\_\_ function.
- (b) Call by reference may be achieved by passing parameter as \_\_\_\_\_.
- (c) A function cannot return more than \_\_\_\_\_ value.
- (d) A program starts execution from \_\_\_\_\_ function.
- (e) Pointer variable is a variable that contains the \_\_\_\_\_ of another variable.
- (f) Elements of an array are stored in \_\_\_\_\_ memory location.

- (iii) What is a structure ? How is a structure declared and how its members are accessed ?
- (iv) In C “ \* ” may be treated as both unary and binary operator. Justify your answer with example.
- (v) Define the range of the random numbers generated by the following expressions :

(a)  $\text{rand}() \% 10 + 1$

(b)  $\text{rand}() \% 2 + 10$

(c)  $\text{rand}() \% 50 - 10$

(vi) `int p=40, q=25, r, s;`

`r = (p++) + (q++);`

`s= (++p) + (++q);`

`printf("%d %d",r,s)`

What will be the output of the above code ?

- (vii) How do we declare variables and symbolic names in a program ? What is the difference between the two ?

- (viii) (a) What functions may be used for input and output of a single character ?
- (b) What is the type of return value of function sizeof() ?
- (ix) What is the difference between while loop and do-while loop ?
- (x) What is the difference between call by value and call by reference ?
- (xi) Write the assignment statements in C for the following :

(a)  $u = y^3 + \log_{10} \left( \frac{y}{i} \right)$

(b)  $v = \frac{1}{2} e^{-\frac{x^2}{2}}$

- (xii) Evaluate the following for int m=2, n=5;

(a)  $++n*n-- + 3*++m$

(b)  $m\%(n*6\%4)$

(c)  $m+++*m*n++$

2. Write the output of any *two* parts from the following :  $2 \times 5 = 10$

```
(i) #include<stdio.h>
main()
{
    int i, j, k, x = 0;
    for(i = 0; i < 5; i++)
        for(j = 0; j < i ; j++)
        {
            switch (i+j-1)
            {
                case -1:
                case 0:
                    x +=1;
                    break;
                case 1:
                case 2:
                case 3:
                    x +=2;
                    break;
                default:
                    x +=3;
            }
            printf("%d", x);
        }
    printf("\n x = %d", x);
}
```

(ii) #include<stdio.h>

```
int sum (int*, int, int*);
```

```
int main (void)
```

```
{
```

```
int a = 4, b=17;
```

```
int c[5] = {9,14,3,15,6};
```

```
int *pc = c;
```

```
a=sum(pc, a, &b);
```

```
printf("2. %d %d %d %d %d %d %d\n",
```

```
a, b, c[0], c[1], c[2], c[3], c[4]);
```

```
return 0;
```

```
}
```

```
int sum (int *px, int y, int *pz)
```

```
{
```

```
int i=5;
```

```
int *p;
```

```
printf("1. %d %d %d\n",*px, y,*pz);
```

```
for (p=px; p<px+5; p++)
```

```
    *p=y+*p;
```

```
    *px=2*i;
```

```
    return(*pz + *px + y);
```

```
}
```

```
(iii) #include<stdio.h>
```

```
int func( int v)
```

```
{ if ( v == 1 || v == 0)
```

```
    return 1;
```

```
    if( v%2 == 0)
```

```
        return func(v/2) + 2;
```

```
    else
```

```
        return func(v-1) + 3;
```

```
}
```

```
main( )
```

```
{
```

```
    printf("%d\n", func(7));
```

```
}
```

3. Attempt any *two* parts :  $2 \times 5 = 10$

(i) Explain the concept of data type FILE. How do you open/close a data file in C ? How do you read or write data in these files ? Why is it important to close the data file ?

(ii) What are function prototypes ? What is their purpose ? Where within a program are function prototypes normally placed ? How are function prototypes declared ?

(iii) #include<stdio.h>

```
void funct(int *p);
```

```
main()
```

```
{ static int x[ 5]= {1,2,3,4,5};
```

```
    funct(x);
```

```
}
```

```
void funct( int *p)
```

```
{
```



```
int i, prod=0;

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

    prod *= *(p+ i);

printf( "product= %d", prod);

return;

}
```

- What type of argument is passed to funct ?
- What value is returned by funct ?
- What information is passed to funct ?
- What is the purpose of the for loop in funct ?
- What is the output of the program ?

4. Attempt any *two* parts :

$$2 \times 4\frac{1}{2} = 9$$

- Develop a C program to draw a random sample of size  $n$  from gamma distribution with parameters  $\lambda$  and  $k$ .
- Write a C program to form a frequency table for marks (integers only) ( $x_i$ ,  $i = 1, 2, \dots, n \leq 100$ ) and  $0 \leq x_i \leq 50$  with an interval of 10.

- (iii) Write a C recursion function to find the value of  $n!$  and use it to compute  ${}^n C_r$  ( $n > r$ ).

5. Attempt any *two* parts : 2×8=16

- (i) Develop a function to calculate correlation coefficient for the data  $(x_p, y_p, i = 1, 2, \dots, n \leq 50)$ . Hence, using the function, develop a program to compute regression line X on Y.

- (ii) Given the data  $(x_p, y_p, i = 1, 2, \dots, n \leq 25)$ , develop a C function program which computes ranks and uses it to calculate Spearman rank correlation between X and Y.

- (iii) Write a C program to fit a Binomial distribution for the given discrete data in the form :  $\{(x_p, f_p), i = 1, 2, 3, \dots, n \leq 25\}$ , and also test for chi-square goodness of fit.