IC-1711

M.Sc. (INFORMATICS) / 1st Semester 2017

PAPER IT-11-Programming Methodology

TIME: 03 hours **Max Marks: 75**

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

Q 1 is compulsory.

Attempt any 4 questions from Q.2 to Q6.

Q1. What do following signifies:

a. const int * const * pcpci;

b. An array G[50][20] is stored in the memory along the row with each of its elements occupying 8 bytes. Find out the location of G[10][15], if G[0][0] is stored at 4200. Also find the total number of elements in the array. (2)

c. Differentiate between int (*daytab)[13] and int *daytab[13]? (3)

d. void * display(int , char **); (2)

e. char (*(*y())[])(); (2)

f. char (*(*x[3])())[5]; (3)

(3)

Q2.

a. Explain all phases through which "C" program passes before being transformed into an executable form? (3)

b. Identify which of following are declarations and which are definitions? (3)

extern int a;	
float z;	Dec
double * fn (int *, void **);	

float square (float x){
//Some code here
}	

c. Write the output of following statements: (3)

printf("%.2s","India");	In
printf("%.2f ", 123.1265);	123.13
printf("New\rDelhi\n");	Delhi

d. What is off-by-one error in looping? Give example? Provide output of following program? (3)

```
#include<stdio.h>
int main(void)
{
            int a=0,b=0;
            if(!a)                a=0  b=1
            {
                        b = !a;
                        if(b)
                                    a = !b;
            }
            printf("%d, %d\n",a,b);
            return 0;
}
```

Scanned by CamScanner

e. What are different scopes of identifiers supported? Provide output of following program? (3)

```
#include<stdio.h>
int main(void)
{
        int i=9;
        if(i==9)
        {
                int i=25;
        }
        printf("i=%d\n",i);
        return 0;
}
```

P B F

Q3.

a. What is Dynamic memory management in C? What are its advantages over static memory allocation? Explain the functions used for dynamic memory management? (5)

b. What are dangling pointers? Write your own version of free function to avoid such problem on dangling pointers? Point out logical error in following statements? (5)

```
int *p1 = (int*) malloc(sizeof(int));
int *p2 = p1;
free(p1);
free(p2);
```

c. What are command line arguments in C? Write sample program to describe how such arguments can be accessed? Provide output of following program? (5)

```
#include<stdio.h>
int main(void)
{
        char *ptr;
        ptr="My name is %s and age is %d\n";
        printf(ptr,"Ranju",30);
        return 0;
}
```

```
Char str[]-" ---
Char tok[]={}
strtok(str, tok);
Char *t
while ( )
{ t= strtok( )
  t++
}
```

Q4.

a. What are Enumerations? Why enumerations are used? (3)

b. Explain different kind of storage classes supported with example? (5)

c. Write down usage of strtok function with example ? Provide the output of following program. (5)

```
#include<stdio.h>
#include<string.h>
int main(void)
{
        printf("Determination"+strlen("Deepali"));
}
```

d. What is use of volatile qualifier? (2)

Q5.

a. What are self-referential structures? Give example. (2)

b. Differentiate between structures and union? Give example. (3)

c. Differentiate between macros and functions? Provide output of following program (Assume 32 bit architecture machine). (5)

```
#include<stdio.h>
#define dp double *
int main(void)
{
        dp p1, p2, p3;
        typedef double *dptr;
        dptr ptr1, ptr2, ptr3;
        printf("%u  %u  %u\n", sizeof(p1), sizeof(p2), sizeof(p3) );
        printf("%u  %u  %u\n", sizeof(ptr1), sizeof(ptr2), sizeof(ptr3) );
        return 0;
}
```

4(8) 4 84

Example [illegible]

488
444

d. Differentiate between function parameters and arguments? (2)

e. Which are recursive functions? Draw stack diagram of factorial calculation using recursion? (3)

Q6.

a. What are Boolean operators? Give example. (2)

13

b. List 3 valid arithmetic operations that can be performed with pointers? (3)

c. Provide output of following: (3)

```
#include<stdio.h>
int main(void)
{
        0 1 2
        int arr[10]= {25,30,35,40,55,60,65,70,85,90},*p;
        for(p=arr+2; p<arr+8; p=p+2)
                printf("%d  ",*p);
        return 0;}
```

① 35, 55, 65 /8

②

③

→ 35 55

65 85

d. Provide output of following program: (4)

```
#include<stdio.h>
int main(void)
{
        int d1,m1,y1;
        char date[11]="24/05/1973";
        date[2]=date[5]='\0';
        sscanf(date,"%d",&d1);
        sscanf(date+3,"%d",&m1);
        sscanf(date+6,"%d",&y1);
        date[2]=date[5]='/';
        printf("d1=%d,m1=%d,y1=%d\n",d1,m1,y1);
        printf("date=%s\n",date);
        return 0;}
```

d1 = 24

m1 = 05

y1 = 1973

date = 24/05/197

e. Provide output of following program: (3)

```
#include<stdio.h>
#include<stdlib.h>
int main(void)
{
            struct  rec
            {
               char *name;
                        int age;
            }*ptr;
            char name[10]="Somalika";
            ptr=(struct rec *)malloc(sizeof(struct rec));
            ptr->name=name;
            ptr->age=93;
            printf("%s\t",ptr->name);
            printf("%d\n",ptr->age);
            return 0;
}
```

Somalika , 93

ato	auto	Register	Static	exter
Storage =	memory	CPU	memory	mem
ef initial value =	unpridictable	garbag	Zero	Zer
pe =	Local	Local	Global	
time =	within Block	within Block	access the function	p

1000

5840