DHARMSINH DESAI UNIVERSITY, NADIAD
FACULTY OF TECHNOLOGY
B.TECH. SEMESTER I [CE/IT]
SUBJECT: (ES105) NAME: PROGRAMMING FOR PROBLEM SOLVING I

Examination	**: First Sessional**	**Seat No**	**: 112**
Date	**: 21/09/2023**	**Day**	**: Thursday**
Time	**: 9:15 a.m. to 10:30 a.m.**	**Max. Marks**	**: 36**

INSTRUCTIONS:

1. Figures to the right indicate maximum marks for that question.
2. The symbols used carry their usual meanings.
3. Assume suitable data, if required & mention them clearly.
4. Draw neat sketches wherever necessary.

Q.1 **Do as directed.** **[12]**

CO1 **R** **(a)** Mention/draw the basic structure of a C program (Brief & Introductory). [2] [2]

CO2 **N** **(b)** What will be the **output/error** in the following codes? [2]

(I)
```
#include<stdio.h>
int main() {
   int 3total, 5total=0;
   3total=1+2+3;
   5total=1+2+3+4+5;
   printf("%d",3total);
   printf("%d",5total);
   return 0;
}
```

(II)
```
#include<stdio.h>
int main()
{
    char volatile='v';
    getchar();
    printf("%c",v);
    return 0;
}
```

CO3 **U** **(c)** What will be the **output/error/warning** in the following code? [2]

```
#include <stdio.h>
int main()
{
    int i=5;
    while(i>0)
    {
        printf("Info i is %d\n",i);
        if(i==3)
            break;
        else
            continue;
        i=i-1;
    }//end of while loop
    printf("Value i is %d\n",i);
    return 0;
}
```

CO1 **R** **(d)** If an integer needs four bytes of storage, then what is the minimum value and maximum value of a signed integer in C? [2]

CO2 **E** **(e)** What will be the **output/error** in the following codes? [2]

(I)
```
#include<stdio.h>
int main() {
   int a = 1;
   if(a--)
      printf("True\n");
   if(++a)
      printf("False\n");
   return 0;
}
```

(II)
```
#include<stdio.h>
int main() {
    int x;
    x = 4<8?5!=1<5==0?1:2:3;
    printf("%d",x);
    return 0;
}
```

CO3 **E** **(f)** What will be the **output/error/warning** in the following code? [2]

```
#include<stdio.h>
int main() {
   int i =0, p = 15;
   while(i<8,p>9) {
      i++;
      p--;
   }
   printf("%d, %d\n", i, p);
   return 0;
}
```

DHARMSINH DESAI UNIVERSITY, NADIAD
FACULTY OF TECHNOLOGY
B.TECH. SEMESTER I [CE/IT]
SUBJECT: (23ES105) PROGRAMMING FOR PROBLEM SOLVING - I

Examination	**: 2nd Sessional**	**Seat No**	**: 112**
Date	**: 26/10/2023**	**Day**	**: Thursday**
Time	**: 11.45am to 01.00pm**	**Max. Marks**	**: 36**

INSTRUCTIONS:
1. Figures to the right indicate maximum marks for that question.
2. The symbols used carry their usual meanings.
3. Assume suitable data, if required & mention them clearly.
4. Draw neat sketches wherever necessary.

Q.1 **Do as directed.** **[12]**

CO1 **R** (a) Describe any two ways to initialize multidimensional array at compile time. [2]

CO1 **U** (b) Summarize the difference between Actual and formal arguments. [2]

CO1 **A** (c) Discover output/error(s) in the following code snippet. [2]

```
#include <stdio.h>
void main()
{
    int a=15;
    int b=10;
    {
        int a=0;
        int c=a+b;
        printf("%d %d\n",b,c);
    }
    b=a;
    printf("%d %d",a,b);
}
```

CO1 **R** (d) Illustrate working of 'atoi' function. [2]

CO4 **N** (e) Differentiate between 'scanf' and 'gets' functions for reading a string from user. [2]

CO5 **E** (f) Judge the following function prototype declarations and provide reason for invalid ones. [2]
1. int (fun) void;
2. double fun (void);
3. fun (int, char, float);
4. float fun (x, y, z);

Q.2 **Attempt *Any TWO* from the following questions.** **[12]**

CO6 **C** (a) Design a C program to search occurrences of element X in given array A, and store all the matching indices of X in Array B. (Note: Assume above program is for integer 1-D array) [6]

CO6 **C** (b) Develop a C program to insert array B inside Array A at given index X. Array A after insertion of Array B at index X should look like following. [6]

Elements of array A	Elements of array B starting from index X	Remaining elements of array A

(Note: Assume above program is for integer 1-D array)

CO6 **E** (c) Evaluate the following code snippets for output/error(s). **[6]**

I)

```
#include<stdio.h>
int main( )
{
 char a[10];
 printf("\nEnter String:");
 scanf("%8s",a);
 printf("%s\n",a);
 printf("%c\n",a[3]+3*2);
 printf("%10s\n",a);
 return 0;
}

// Assume Input string to
// a = "HELLO How"
```

II)

```
#include<stdio.h>
int func(int a, int b);
void main()
{
    int v;
    v=func(3,3);
    printf("%d",v);
}
 int func(int a, int b)
{
    if(b==0)
        return 0;
    if(b==1)
        return a;
    return a + func(a,b-1);
}
```

stdlib.h
abs ⇒ Absolute value

Q.3 **Do as directed.** **[12]**

CO4 **N** (a) Illustrate a C program to compare two strings case insensitively. [6]

CO5 **E** (b) Judge and discuss recursive approach to find factorial of given number. [6]

OR

Q.3 **Do as directed.** **[12]**

CO4 **N** (a) Relate any 3 in-built string functions from 'string.h' header file by stating appropriate example and details. [6]

CO5 **E** (b) Formulate a C program having user defined function to calculate the transpose of a given matrix. [6]

Blooms Taxonomy levels: R-Remembering, U- Understanding, A-Applying, N-Analyzing, E- Evaluating, C-Creating

DHARMSINH DESAI UNIVERSITY, NADIAD
FACULTY OF TECHNOLOGY
B.TECH. SEMESTER I [CE/IT]
SUBJECT: (ES105) NAME: PROGRAMMING FOR PROBLEM SOLVING-I

Examination	: Third Sessional	**Seat No**	: 112
Date	: 20/12/2023	**Day**	: Wednesday
Time	: 9:15 a.m. to 10:30 a.m.	**Max. Marks**	: 36

INSTRUCTIONS:

1. Figures to the right indicate maximum marks for that question.
2. The symbols used carry their usual meanings.
3. Assume suitable data, if required & mention them clearly.
4. Draw neat sketches wherever necessary.

Q.1 Do as directed. [12]

CO2 U **(a)** Interpret and differentiate the given declarations: [2]

(1) int *(*p[4])(int*, int*);
(2) int *(*q)();

CO5 E **(b)** What will be the **output/error/warning** in the following codes? [2]

(I)
```
#include<stdio.h>
void foo(char* a) {
   if(*a && *a != ' '){
         foo(a+1);
         putchar(*a);
   }
}
int main() {
   foo("ABCD EFGH");
   return 0;
}
```

(II)
```
#include<stdio.h>
#include<string.h>
int main()
{
 char c[] = "DDUBTECH2023";
 char* p = c;
 printf("%d",
    (int)strlen(c+7[p]-3[p]+2));
 return 0;
}
```

CO5 E **(c)** What will be the **output/error/warning** in the following code? [2]

Note: **The fmax() family of functions determine the maximum numeric value of their arguments.**

```
#include <stdio.h>
#include <math.h>
int f(int* p, int n){
 if(n<=1)
    return 0;
 else
    return fmax( f(p+1,n-1), p[0] - p[1]);
}
int main(){
    int a[] = {3,5,2,6,4};
    printf("%d",f(a,5));
    return 0;
}
```

CO1 R **(d)** What is the purpose of using **typedef** with structure definition? Explain with example. [2]

CO1 A **(e)** Assume that the size of the data types short, float, and long occupy 2 bytes, 4 bytes, and 8 bytes respectively. What will be the memory requirement for variable stv, ignoring alignment considerations? [2]

```
struct{
  short sha[5];
  union{
       float fly;
       long lnz;
  }un;
}stv;
```

CO3 A **(f)** Write a code to compare two variables of the below user-defined data type: [2]

```
struct student{
   char name[30];
   int rollNo;
```

int marks;
};

Q.2 Attempt ***Any TWO*** from the following: [12]

CO2 U (a) (I) Explain Pointer to Pointer(Chain of Pointers) with suitable code as an example. [6]
(II) Explain your understanding about scaling factor of pointers to primitive datatypes.

CO4 N (b) With reference to relation between 2-D arrays and pointers, answer following: [6]
Given int arr[m][n] be a 2-D array. For following p, q, r different types of pointers, derive pointer notations to access element a[i][j]:
(I) int* p = &arr[0][0];
(II) int (*q)[n] = arr;
(III) int (*r)[m][n] = &arr;

CO2 U (c) From the list of operation involving pointers(p and q are pointing to different locations of same array) as below, mention which are supported and why. Also, which are not supported with reasoning: [6]
(I) p / 4 (II) p + 5.6 (III) p - q (IV) p + q (V) p * q (VI) p * 4

Q.3 Attempt the following:

CO6 C (a) Design and develop complete C program to meet following requirements: [12]
For an event, the organizer is required to collect information of invited guests. Each invited guest has to provide his full name and count of family members including him who are going to attend this event.

→ Define struct guest containing members name and familyCount.

Write user-defined function to collect this information

→ **void collectGuestsInfomation(int n, struct guest guestsInfo[]);**

Also, organizer is required to maintain internally the list of guests sorted in ascending order according to the family count. Note that the memory of original guestsInfo array must not be disturbted/altered.

Use array of pointers **'agp'** in main function to initially point to each element within array of structures.

→ **void sortGuests(int n, struct guest **agp); //must use pointer notation to processing any elements**

Utilize all user defined functions using the main function to complete the job. Display two outputs - guests information as per input ordering. - sorted guests information as per their count of family members.

OR

Q.3 Attempt the following:

CO6 C (a) Design and develop complete C program to meet following requirements: [12]
For a ecommerce company, the manager is required to collect information of customers. Each customer's full name and total spending in rupees so far is required to be stored as tha data.

Define struct customer containing customer's name and totalSpending.

Write user-defined function to store this information
void storeCustomersInfomation(int n, struct customer customersInfo[]);

Also, company is planning to give some offers/coupans to special customers who are having maximum spending so far.

Note that while figuring out list of special customers, do not duplicate as well as modify/alternate customers data in the memory stored.

Write a single function to find maximum spending value first.

void findMaxSpending(int n, struct customer customersInfo[], int *pmaxSpendingValue);

Write another function to save pointers of those customers only whose spending matches to maxSpendingValue.

Use array of pointers **'arrcustp'** in main function which is initially garbage when passed but upon return it has information of all special customers. The number of such customers is returned using return keyword by the below function.

int findMaxSpendingCustomers(int n, struct customer customersInfo[], struct customer **arrcustp, int maxSpendingValue);

Utilize all user defined functions using the main function to complete the job. Display special customers's information.

* Bloom's Taxonomy levels : R-Remembering, U- Understanding, A-Applying, N-Analyzing, E- Evaluating, C-Creating