**MST -3**

**2nd SEMESTER**

**CS101-Introduction to C Programming**

**Set- B**

**Time allowed: 120 Minutes Max. Marks: 30**

**General Instructions:**

* **All Questions are mandatory**

1. Predict the output of the following C program

#include<stdio.h>

struct Point

{

int x, y, z;

};

int main()

{

struct Point p1 = {.y = 0, .z = 1, .x = 2};

printf("%d %d %d", p1.x, p1.y, p1.z);

return 0;

}

1. **2 0 1**
2. 2 1 0
3. 0 1 2
4. Compiler error
5. Predict the output of the following C code:

#include <stdio.h>

enum day {newton=4, turing};

int main()

{

printf("%d", turing);

return 0;

}

1. 1
2. Error
3. **5**
4. 0

3. What is the benefit of structure?

A) It is used to define primitive data type which can store heterogeneous data

B) It is used to define primitive data type which can store homogeneous data

**C) It is used to define non-primitive data type which can store heterogeneous data**

D) It is used to define non-primitive data type which can store homogeneous data

1. Which among the following is never possible in C when members in a structure are same as that in a union?

//Let P be a structure

//Let Q be a union

a. sizeof(P) is greater than sizeof(Q)

b. sizeof(P) is equal to sizeof(Q)

**c. sizeof(P) is less than to sizeof(Q)**

d. None of the above

5. The correct syntax to use typedef for struct is.

a) typedef struct temp

{

int a;

}TEMP;

b) typedef struct

{

int a;

}TEMP;

c)struct temp

{

int a;

};

typedef struct temp TEMP;

**d)All of the mentioned**

1. Predict the output of the C code. Assume the size of integer is 4 bytes and size of character is 1 byte.

union test

{

int x;

char arr[8];

int y;

};

int main()

{

printf("%d", sizeof(union test));

return 0;

}

1. 12
2. **8**
3. 16
4. Compiler error
5. Predict the output of following code.

#include<stdio.h>

struct st

{

int y;

};

struct st1

{

int x;

struct st next;

};

int main()

{

struct st1 temp;

int k=temp.x = 10;

temp.next.y = k;

printf("%d", temp.next.y);

return 0;

}

1. **10**
2. Compiler error
3. Run time Error
4. Junk
5. What will the output of following code.

#include <stdio.h>

struct student

{

char \*name;

};

struct student s;

struct student fun(void)

{

s.name = "newton";

printf("%s ", s.name);

s.name = "Edison";

return s;

}

int main()

{

struct student m = fun();

printf("%s ", m.name);

m.name = "turing";

printf("%s ", s.name);

return 0;

}

1. **Newton** **Edison** **Edison**
2. Newton Turing Edison
3. Newton Newton Newton
4. Newton Turing Turing
5. What will be the output of the program (sample.c) given below if it is executed from the command line?

**cmd> *sample one two three***

/\* sample.c \*/

#include<stdio.h>

#include<string.h>

int main(int argc, char \*argv[])  
{  
    int i=0;  
    i+=strlen(argv[1]);  
    while(i>0)  
    {  
        printf("%c", argv[1][--i]);  
    }  
   return 0;  
}

* 1. three two one
  2. owt
  3. **eno**
  4. eerht

1. Predict the output of following code.

#include <stdio.h>

int main()

{

struct student

{

int no;

char name[20];

};

struct student s;

no = 8;

printf("%d", no);

}

* 1. 8
  2. **Compile Time Error**
  3. Run Time Error
  4. Junk

**Coding-1 (5 marks)**

Write a program to store and display the educational details of a particular student. The following details should store: student name, **year of joining and cgpa. The cgpa is to be stored for all the years. If the student is in 3rd year then two cgpas will be stored (one for first year and one for second year).**

**The program should display the details as follows: student name, year of joining,** all cgpa’s individually, total cgpa and average of cgpa (**upto 2 places of decimal).**

Sample Input1:

2//student in 3rd year and has completed 2 years, so 2 cgpa’s will be entered

Rahul

2014

7.896

8.65

Sample Output1:

Rahul

2014

7.90//cgpa first year

8.65//cgpa second year

16.55//

8.27

**Explanation:**

**Sample Input**: First line contains the number of completed years, second lines scans the name of a student and third line represents student’s year of joining. In next lines cgpa of a student is taken as input based on number of years entered in first line.

**Sample Output**: Displays the details of student. First line is the name of student that was entered, next line contains year of joining and next lines displays the cgpas entered based on number of years. Second last line gives total of all cgpas and last line displays the average (upto 2 decimal places).

Sample Input1:

2//student in 3rd year and has completed 2 years, so 2 cgpa’s will be entered

Rahul

2014

7.896

8.65

Sample Output1:

Rahul

2014

7.90//cgpa first year

8.65//cgpa second year

16.55//

8.27

Input Testcase 1:

4

Suman

2011

8.6

7.5

9.456

8.999

Output Testcase 1:

Suman

2011

8.60

7.50

9.46

9.00

34.56

8.64

Input testcase 2:

3

Surbhi

2015

6.77

8.77

6.54

Output Testcase2:

Surbhi

2015

6.77

8.77

6.54

22.08

7.36

Input testcase 3:

1

Ranvijay

2014

9.456

Output testcase 3:

Ranvijay

2014

9.46

9.46

9.46

Solution-

**<readonly>**

**#include<stdio.h>**

**struct student**

**{**

**char name[30];**

**int yearOfJoining;**

**float cgpa[4];**

**float cgpa\_Total;**

**};**

**void getDetails(struct student\*,int);**

**void displayDetails(struct student\*,int);**

**void calculateCGPA(struct student\*,int);**

**int main()**

**{**

**int n;**

**struct student stu,\*ptr\_stu;**

**ptr\_stu=&stu;**

**scanf("%d",&n);    //number of completed years**

**getDetails(ptr\_stu,n);**

**calculateCGPA(ptr\_stu,n);**

**return 0;**

**}**

**</readonly>**

void getDetails(struct student \*ss,int n)

{

int i;

scanf("%s",ss->name);

scanf("%d",&ss->yearOfJoining);

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

{

scanf("%f",&ss->cgpa[i]);

}

}

void calculateCGPA(struct student \*stud,int n)

{

int i;

float avg=0.0;

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

{

stud->cgpa\_Total+=stud->cgpa[i];

}

displayDetails(stud,n);

printf("%.2f\n",stud->cgpa\_Total);

avg=stud->cgpa\_Total/n;

printf("%.2f",avg);

}

void displayDetails(struct student \*s,int n)

{

int i;

printf("%s\n",s->name);

printf("%d\n",s->yearOfJoining);

for(i=0;i<n;i++){

printf("%.2f\n",s->cgpa[i]);

}

}

**Coding-2(10 marks)**

Alisha has a list of N students of CSE branch containing Roll No., Name and Marks . She wants to display the name of the topper of the class.

If there are more than one student with same marks then display the name of the student whose name is coming alphabetically first.

**Sample Testcase:**

**Input**

3 //Number of students

101 //Rollno

Rina //Name

90 //Marks

102

Heena

78

103

Karan

90

**Output**

Karan

Explanation: Karan and Rina both have same name but name of "Karan" comes first alphabetically.

Write a C Program to help her.

Constraint:

1<=N<=10

**SampleInput**

3

101

karan

45

102

Jatin

60

103

Tanmay

75

**Sample Output**

Tanmay

Testcase 1:

Input:

4

10

Karan

90

11

Jatin

60

12

Tanmay

75

13

Damini

67

Output:

Karan

TestCase 2:

Input:

4

101

Karan

80

102

Jatin

70

103

Tanmay

75

104

Kirti

80

Output:

Karan

Testcase 3:

5

101

Karan

80

102

Jatin

93

103

Tanmay

75

104

Kirti

98

105

Nidhu

89

**Output:**

Kirti

**Testcase 4:**

5

101

Karan

80

102

Jatin

93

103

Tanmay

75

104

Kirti

90

105

Nidhu

89

**Output:**

Jatin

Testcase 5:

10

1

Heena

67

2

Tanmay

68

3

Saanvi

89

4

Nidhu

89

5

Karan

80

6

Saanvi

45

7

Suhana

67

8

Namish

93

9

Nidhu

98

10

Heena

98

Output:

Heena

**Code Stub**

**#include <math.h>**

**#include <stdio.h>**

**#include <string.h>**

**#include <stdlib.h>**

**#include <assert.h>**

**#include <limits.h>**

**#include <stdbool.h>**

**struct student**

**{**

**int r;**

**char name[50];**

**int marks;**

**};**

**int main()**

**{**

**struct student s[10];**

**int n,i,max;**

**char temp[50];**

**scanf("%d",&n);**

**//Write your Code Here**

**Solution**

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

{

scanf("%d%s%d",&s[i].r,s[i].name,&s[i].marks);

}

max = s[0].marks;

strcpy(temp,s[0].name);

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

{

if(s[i].marks>max)

{

max=s[i].marks;

strcpy(temp,s[i].name);

}

else

if(s[i].marks==max && (strcmp(temp,s[i].name)>0))

{

max=s[i].marks;

strcpy(temp,s[i].name);}

}

printf("%s",temp);

return 0;

}