**FA-4 Team3 set 2**

**4 Mcqs of 1 mark each**

Que1. Size of a union is determined by size of the.  
a) First member in the union  
b) Last member in the union  
**c) Biggest member in the union**d) Sum of the sizes of all members

Que 2 What is the output of this C code?

#include <stdio.h>

struct student

{

char \*c;

};

int main()

{

struct student \*s;

s->c = "hello";

printf("%s", s->c);

}

a) **hello**   
b) Segmentation fault  
c) Run time error  
d) Nothing

|  |  |  |
| --- | --- | --- |
| Que 3 What is the similarity between a structure, union and enumeration?   |  |  | | --- | --- | | a) | All of them let you define new values | |
| |  |  | | --- | --- | |  |  | | **b)** | **All of them let you define new data types** | | c) | All of them let you define new pointers | | d) | All of them let you define new structures | |

Que 4.  Which of the following reduces the size of a structure?  
a) union  
**b) bit-fields**c) malloc  
d) none of the mentioned

**3 Mcqs of 2 mark each**

**Que 5** What is the output of following program?

|  |
| --- |
| #include <stdio.h>  struct student  {  char \*name;  };  struct student fun(void)  {  struct student s;  s.name = "newton";  return s;  }  void main()  {  struct student m = fun();  s.name = "turing";  printf("%s", m.name);  } |

1. turing
2. newton
3. **compile time error**
4. turing newton

Que 6 Predict output of following

#include<stdio.h>

int main()

{

    struct site

    {

        char name[] = "C program";

        int no\_of\_pages = 200;

    };

    struct site \*ptr;

    printf("%d ", ptr->no\_of\_pages);

    printf("%s", ptr->name);

   return 0;

}

1. 200 C program
2. 200
3. **Compile time error**
4. Run time error

Que 7 What is the output of this C code?

#include <stdio.h>

struct p

{

int k;

char c;

float f;

};

int p = 10;

int main()

{

struct p x = {1, 97};

printf("%f %d**\**n", x.f, p);

}

a) Compile time error  
b**) 0.000000 10**c) Somegarbage value 10  
d) 0 10

**Coding Question-10 marks**

Ques -Write a program to help a teacher to find the student who scores are maximum and minimum marks in the exam.

**Sample Input:**

5 //number of students  
1100  
Ram  
87  
1101  
Mohit  
32  
1102  
Kavish  
98  
1103  
Ravi  
65  
1104  
Arun  
22

**Explanation:** In sample input first number represent the number of student in class. and then it accepts three input more for each student.

First input represents the roll number of the student.

Second input represents the name of the student.

Third input represents the marks of the student.

**Sample Output:**

1102  
Kavish  
98  
1104  
Arun  
22

**Explanation:** Sample output include six input for two student. One student who score maximum marks with details that include its roll number, name and marks and one student who score minimum marks with details that include its roll number, name and marks.

Input Testcase 1:

5

101

Ram

98

102

Shyam

54

103

Rohan

67

104

Sohan

53

105

Rahul

65

Output testcase 1:

101

Ram

98

104

Sohan

53

Input testcase2:

10

100

Z

89

99

Y

65

98

X

56

97

W

72

96

V

27

95

U

99

94

T

69

93

S

75

92

R

12

91

Q

49

Output Testcase 2:

95

U

99

92

R

12

Input testcase3:

7

16001

Abc

22

16002

Def

77

16003

Ghi

19

16004

Jkl

65

16005

Mno

54

16006

Pqr

87

16007

Stu

33

Output testcase 3:

16006

Pqr

87

16003

Ghi

19

Input testcase 4:

2

102

Shyam

40

101

Ram

98

Output testcase 4:

101

Ram

98

102

Shyam

40

Input testcase 5:

3

1

Alam

65

2

Dev

87

3

Marc

54

Output Testcase 5:

2

Dev

87

3

Marc

54

**#include <math.h> //read only code stub**

**#include <stdio.h>**

**#include <string.h>**

**typedef struct**

**{**

**char name[10];**

**int marks;**

**int roll;**

**}student;**

**int last(student [],int);**

**int first(student [],int);**

**int main()**

**{**

**student stu[50];**

**int i=0;**

**int n1,n2,h1,h2,m;**

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

**for(i=0;i<m;i++)**

**{**

**scanf("%d",&stu[i].roll);**

**scanf("%s",stu[i].name);**

**scanf("%d",&stu[i].marks);**

**}**

**int f=first(stu,m);**

**printf("%d\n%s\n%d\n",stu[f].roll,stu[f].name,stu[f].marks);**

**int s=last(stu,m);**

**printf("%d\n%s\n%d",stu[s].roll,stu[s].name,stu[s].marks);**

**return 0;**

**}**

**Solution**-

int last(student stu[], int m)

{

int i,l=stu[0].marks,index=0;

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

{

if(stu[i].marks<l)

{

l=stu[i].marks;

index=i;

}

}

return index;

}

int first(student stu[], int m)

{

int i,f=stu[0].marks,index=0;

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

{

if(stu[i].marks>f)

{

f=stu[i].marks;

index=i;

}

}

return index;

}