Course: C **Session:** Structures and Unions **Timestamp:** 2021-1-9 19:41:44 **Register Number:** RA2031241010094

Q. Sum of cube of Digits

Ganesh played game with his friend sanjay to find Sum of Cube of digits of the number he said. Help to Ganesh and sanjay to solve it by your code using union.

Input Method

Integer ranges from 1 to 999

Output Method

Sum of Cube of digits of the number

Source Code

```
#include <stdio.h>
int main()
{
  int i,n,dig,sum=0,cube=0;
  scanf("%d",&n);
  while(n>0)
  {
    dig=n%10;
    sum=sum+dig;
    n=n/10;
  }
  cube=cube+sum*sum*sum;
 printf("Sum=%d\n",sum);
  printf("Cube=%d",cube);
 return 0;
}
```

Sample Input

123

Sample Output

Sum=6
Cube=216

Result

Thus, Program " **Sum of cube of Digits** " has been successfully executed

Q. Time Machine Difference

The Time Machine is a science fiction novella by H. G. Wells, published in 1895 and written as a frame narrative. The work is generally credited with the popularization of the concept of time travel by using a vehicle that allows an operator to travel purposely and selectively forwards or backwards in time. The term "time machine", coined by Wells, is now almost universally used to refer to such a vehicle.

The Time Machine has been adapted into three feature films of the same name, as well as two television versions, and a large number of comic book adaptations. It has also indirectly inspired many more works of fiction in many media productions. Now kindly help our Captain of the Ship "H.G.Wells" to calculate the difference between the two time machines.

Mandatory:

1. Create a Structure "Time" and three data members as seconds(int), minutes(int), hours(int)

2. The Structure Variables as "startTime, stopTime, diff"

3.Access the data members as follows

stopTime.seconds
startTime.hours

Note: The structure variables, data members and structure name are CASE Sensitive.

Follow the same case mentioned in the mandatory

Source Code

```
#include <stdio.h>
struct time
{
  int hours,minutes,seconds;
};
void difference(struct time t1,struct time t2,struct time*diff);
int main()
{
  struct time startTime,stopTime,diff;
  scanf("%d %d %d",&startTime.hours,&startTime.minutes,&startTime.seconds);
  scanf("%d %d %d",&stopTime.hours,&stopTime.minutes,&stopTime.seconds);
  difference(startTime,stopTime,&diff);
  printf("TIME DIFFERENCE=%d:%d:%d-",startTime.hours,startTime.minutes,startTime.seconds);
  printf("%d:%d:%d=",stopTime.hours,stopTime.minutes,stopTime.seconds);
  printf("%d:%d:%d",diff.hours,diff.minutes,diff.seconds);
 return 0;
}
void difference(struct time start,struct time stop,struct time*diff)
{
  if(stop.seconds>start.seconds)
  {
    --start.minutes;
    start.seconds+=60;
  }
  diff->seconds=start.seconds-stop.seconds;
  if(stop.minutes>stop.minutes)
  {
    --start.hours;
    start.minutes+=60;
  }
  diff->minutes=start.minutes-stop.minutes;
  diff->hours=start.hours-stop.hours;
}
```

Sample Input

```
12 34 55
8 12 15
```

Sample Output

```
TIME DIFFERENCE=12:34:55-8:12:15=4:22:40
```

Result

Thus, Program " **Time Machine Difference** " has been successfully executed

Course: C **Session:** Structures and Unions **Timestamp:** 2021-1-9 19:42:46 **Register Number:** RA2031241010094

Q. Sum of Digits

Jagan played maths game with his friend sanjay to find sum of digits of the number he said.
Help to jagan and sanjay to solve it by your code using union.

Input Method

Integer ranges from 1 to 999

Output Method

sum of digits of the number

Mandatory:

1. Create union name as "Data"
2. The variable name for union is "data"

Source Code

```
#include <stdio.h>
int main()
{
  int i,data,sum=0,dig;
  scanf("%d",&data);
   while(data>0)
   {
     dig=data%10;
     sum=sum+dig;
     data=data/10;
   }
   printf("%d",sum);
 return 0;
}
```

Sample Input

123

Sample Output

6

Result

Thus, Program " **Sum of Digits** " has been successfully executed

Course: C **Session:** Structures and Unions **Timestamp:** 2021-1-9 19:43:40 **Register Number:** RA2031241010094

Q. Payroll using Structures

1. Create a Structure "employee"
2. Create six data members for structures as name(char), empid(int), salary(int), hra(int), da(int), total(float)
3. Input the data of the employee as name, empid, salary.
4. Calculate the HRA(10% salary), DA(20% salary)
5. Total pay = salary +hra +da
6. Create structure variable as "emp"

Source Code

```
#include <stdio.h>
struct employee
{
  char name[20];
  int empid,salary,hra,da;
  float total;
}emp;
int main()
{
  scanf("%s",emp.name);
  scanf("%d\n%d",&emp.empid,&emp.salary);
  emp.hra=0.1*emp.salary;
  emp.da=0.2*emp.salary;
  printf("Name=%s\n",emp.name);
  printf("Id=%d\n",emp.empid);
  printf("HRA=%d\n",emp.hra);
  printf("DA=%d\n",emp.da);
  printf("Total Salary=%.0f",emp.salary+(emp.salary*.1)+(emp.salary*.2));
 return 0;
}
```

Sample Input

```
Bogar
1000
15000
```

Sample Output

```
Name=Bogar
Id=1000
HRA=1500
DA=3000
Total Salary=19500
```

Result

Thus, Program " **Payroll using Structures** " has been successfully executed

Q. Student Management Array of Structure

1 Create a structure "student".

2. This structure has three members: name (string), roll (integer) and marks (float).

3. Created a structure array of size 3 to store information of 3 students and structure variable as "s".

Hint: s[3];

4. Using for loop, the program takes the information of 3 students from the user and displays it on the screen.

Source Code

```
#include <stdio.h>
struct stident
{
  char name[10];
  int roll;
  float marks;
}s[3];
int main()
{
  int i;
  for(i=0;i<3;i++)
  {
    s[i].roll=i+1;
    scanf("%s",s[i].name);
    scanf("%f",&s[i].marks);
  }
  for(i=0;i<3;i++)
  {
    printf("Roll number=%d\n",i+1);
    printf("Name=%s\n",s[i].name);
 printf("Marks=%.2f\n",s[i].marks);
  }
 return 0;
}
```

Sample Input

```
Bogar 99.33
Siddhar 99.44
Tamil 99.99
```

Sample Output

```
Roll number=1
Name=Bogar
Marks=99.33
Roll number=2
Name=Siddhar
Marks=99.44
Roll number=3
Name=Tamil
Marks=99.99
```

Result

Thus, Program " **Student Management Array of Structure** " has been successfully executed

Course: C **Session:** Structures and Unions **Timestamp:** 2021-1-9 19:46:01 **Register Number:** RA2031241010094

Q. TRANSFORMATION

Reverse Polish Notation (RPN) is a mathematical notation where every operator follows all of its operands. For instance, to add three and four, one would write "3 4 +" rather than "3 + 4". If there are multiple operations, the operator is given immediately after its second operand; so the expression written "3 4 + 5" would be written "3 4 5 +" first subtract 4 from 3, then add 5 to that.
Transform the algebraic expression with brackets into RPN form.
You can assume that for the test cases below only single letters will be used, brackets [] will not be used and each expression has only one RPN form (no expressions like a*b*c)
Input
The first line contains t, the number of test cases (less then 100).
Followed by t lines, containing an expression to be translated to RPN form, where the length of the expression is less then 400.
Output
The expressions in RPN form, one per line.
Example
Input:
3
(a+(b*c))
((a+b)*(z+x))
((a+t)*((b+(a+c))^(c+d)))

Output:
abc*+
ab+zx+*
at+bac++cd+^*

Source Code

```
#include <stdio.h>
int main()
{
  int t,top=0,i;
  char exp[400],stack[200];
  scanf("%d",&t);
  while(t--)
  {
    scanf("%s",exp);
    for(i=0;exp[i];i++)
    {
      if(exp[i]=='(')
        continue;
      else if(exp[i]==')')
      {
        printf("%c",stack[top]);
        top--;
      }
      else if(exp[i]=='+'||exp[i]=='-'||exp[i]=='*'||exp[i]=='/'||exp[i]=='^')
      {
        top++;
        stack[top]=exp[i];
      }
      else
      {
        printf("%c",exp[i]);
      }
    }
    printf("\n");
  }
 return 0;
}
```

Sample Input

```
3
(a+(b*c))
((a+b)*(z+y))
((a+t)*((b+(a+c))^(c+d)))
```

Sample Output

```
abc*+
ab+zy+*
at+bac++cd+^*
```

Result

Thus, Program " **TRANSFORMATION** " has been successfully executed

Q. Structure 42

Write a program , using a pointer to a structure to initialize the members of the structure to display the students course registration details where details of first student initialized in the program and details of second student get from the user , then display the details of both student 1 and student 2

Input and Output Format:

Refer sample input and output for formatting specification.

All float values are displayed correct to 2 decimal places.

All text in bold corresponds to input and the rest corresponds to output.

Note:

Use structure and union concepts

Source Code

```
#include <stdio.h>
struct student
{
  int roll,fee;
  char name[10],dp[5];
}s;
int main()
{
  scanf("%d",&s.roll);
  scanf("%s",s.name);
  scanf("%s",s.dp);
  scanf("%d",&s.fee);
  printf("Roll no:%d\n",s.roll);
  printf("Name:%s\n",s.name);
  printf("Course:%s\n",s.dp);
  printf("Fees:%d\n",s.fee);
 return 0;
}
```

Sample Input

```
12
ram
it
2333
```

Sample Output

```
Roll no:12
Name:ram
Course:it
Fees:2333
```

Result

Thus, Program " **Structure 42** " has been successfully executed

Course: C **Session:** Structures and Unions **Timestamp:** 2021-1-9 19:46:59 **Register Number:** RA2031241010094

Q. Time Difference

Help Rama to achieve his friends task "Time challenge" to display hours minutes and seconds in both 12 and 24 hours format: 24 Hours format : 23:30:12 Standard format : 11:30:12 pm
Refer the sample input and output:

Sample Input 1:
20 25 06

Output 1:
24 Hours Format
20:25:06
12 Hours Format
08:25:06 pm

Sample Input 2:
24 25 06

Output 2:
Invalid Time

Sample Input 3:
23 60 06

Output 3:
Invalid Time

Sample Input 4:
23 56 61

Output 4:
Invalid Time

Source Code

```
#include <stdio.h>
struct time
{
  int h,m,s;
}a;
int main()
{
  scanf("%d %d %d",&a.h,&a.m,&a.s);
  if(a.h>=24&&a.m>1&&a.s>2)
  {
    printf("Invalid Time");
  }
  else if(a.m>60||a.s>60)
  {
    if(a.s>60||a.m>60)
    {
    printf("Invalid Time");
    }
  }
  else if(a.h<=12)
  {
  printf("24 Hours Format\n");
  printf("%d:%d:%d\n",a.h,a.m,a.s);
  printf("12 Hours Format\n");
  printf("%d:%d:%d am",a.h,a.m,a.s);
}
  else
  {
    printf("24 Hours Format\n");
    printf("%d:%d:%d\n",a.h,a.m,a.s);
    printf("12 Hours Format\n");
 printf("%d:%d:%d pm",a.h-12,a.m,a.s);
  }
 return 0;
}
```

Sample Input

```
23
35
22
```

Sample Output

```
24 Hours Format
23:35:22
12 Hours Format
11:35:22 pm
```

Result

Thus, Program " **Time Difference** " has been successfully executed

Course: C **Session:** Structures and Unions **Timestamp:** 2021-1-9 19:47:27 **Register Number:** RA2031241010094

Q. Structure 41

Write a program to read display add and subtract of two time variables defined using hours , minutes and seconds using typdef and passing structure variable as argument in functions.
Input and Output Format:
Refer sample input and output for formatting specification

Source Code

```
#include <stdio.h>
struct time
{
  int h,m,s;
};
void add(int,int,int,int,int,int);
int main()
{
  struct time t1,t2;
  scanf("%d %d %d",&t1.h,&t1.m,&t1.s);
  scanf("%d %d %d",&t2.h,&t2.m,&t2.s);
  add(t1.h,t1.m,t1.s,t2.h,t2.m,t2.s);
 return 0;
}
void add(int x,int y,int z,int a,int b,int c)
{
  int ho,mi,se;
  ho=x+a;
  mi=y+b;
  se=z+c;
  printf("%d hrs\n",ho);
  printf("%d min\n",mi);
  printf("%d sec\n",se);
}
```

Sample Input

```
12 12 33
3 1 3
```

Sample Output

```
15 hrs
13 min
36 sec
```

Result

Thus, Program " **Structure 41** " has been successfully executed

Course: C **Session:** Structures and Unions **Timestamp:** 2021-1-9 19:47:52 **Register Number:** RA2031241010094

Q. Printing next 5 numbers

A new game was introduced in a school for students of 3 standard. In which the student should tel next 5 numbers sequence from the telling number. Using union help to the students to solve it.

Input Method

Integer ranges from 1 to 999

Output Method

Sequence of next 5 numbers

Mandatory:

Use union concept

Source Code

```
#include <stdio.h>
union st
{
  int a;
}s;
int main()
{
  int i;
  scanf("%d",&s.a);
  for(i=s.a+1;i<=s.a+5;i++)
 printf("%d ",i);
 return 0;
}
```

Sample Input

8

Sample Output

9 10 11 12 13

Result

Thus, Program " **Printing next 5 numbers** " has been successfully executed