

/\*

Name : Deepanshu Gupta

Section : AI & ML

Roll Number : 10

1. An electronics shop has announced the following seasonal discounts on the purchase of certain items:

PURCHASE AMOUNT IN RS.		DISCOUNT ON LAPTOP	
DISCOUNT ON DESKTOP PC			
0 – 25000	0.0%	5.0%	
25001 – 57000	5.0%		7.6%
57001 – 100000	7.5%	10%	
More than 100000	10.0%	15.0%	

Write a program based on the above criteria to input name, address, amount of purchase and the type of purchase (L for Laptop and D for Desktop) by a customer. Compute and print the net amount to be paid by a customer along with his name and address. (Hint: Discount = (discount rate / 100) \* amount of purchase, Net amount = amount of purchase – discount)

\*/

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    string name, add;
```

```
    char type;
```

```
    float amt, dis_rate, dis, net_amt;
```

```
    cout<<"Enter name: ";
```

```
    getline(cin, name);
```

```
    cout<<"Enter address: ";
```

```
    getline(cin, add);
```

```
    cout<<"Enter amount of purchase: ";
```

```
    cin>>amt;
```

```
    cout<<"Enter type of purchase: ";
```

```
    cin>>type;
```

```
    switch (type)
```

```
{
```

```

case 'L':
    if(amt<=25000)
        dis_rate=0;
    else if(amt>25000 && amt<=57000)
        dis_rate=5;
    else if(amt>57000 && amt<=100000)
        dis_rate=7.5;
    else
        dis_rate=10;
    break;
case 'D':
    if(amt<=25000)
        dis_rate=5;
    else if(amt>25000 && amt<=57000)
        dis_rate=7.6;
    else if(amt>57000 && amt<=100000)
        dis_rate=10;
    else
        dis_rate=15;
    break;
default:
    cout<<"Invalid Type of Purchase\n";
    break;
}
dis=(dis_rate/100)*amt;
net_amt=amt-dis;
cout<<"-----";
cout<<"\nName: "<<name;
cout<<"\nAddress: "<<add;
cout<<"\nNet Amount: "<<net_amt;
return 0;
}

```

\*\*\*\*\*OUTPUT-1\*\*\*\*\*

Enter name: Deepanshu

Enter address: Delhi

Enter amount of purchase: 50000

Enter type of purchase: L

-----

Name: Deepanshu

Address: Delhi

Net Amount: 47500

\*\*\*\*\*OUTPUT-2\*\*\*\*\*

Enter name: Ansh

Enter address: Dehradun

Enter amount of purchase: 75000

Enter type of purchase: D

-----

Name: Ansh

Address: Dehradun

Net Amount: 67500

/\*

Name : Deepanshu Gupta

Section : AI & ML

Roll Number : 10

2. Using the switch statement, write a menu-driven CO1 program to calculate the maturity amount of a bank deposit. The user is given the following options: (i) Term Deposit (ii) Recurring Deposit For option (i) accept Principal (p), rate of interest (r) and time period in years (n). Calculate and output the maturity amount (a) receivable using the formula  $a = p[1 + r/100]^n$ . For option (ii) accept monthly installment (p), rate of interest (r) and time period in months (n). Calculate and output the maturity amount (a) receivable using the formula  $a = p*n + p*n(n+1)/2*r/100*1/12$ . For an incorrect option, an appropriate error message should be displayed

\*/

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int type;
```

```
    float p, r, n, a;
```

```
    cout<<"1. Term Deposit\n2. Recurring Deposit\n";
```

```
    cout<<"Enter your choice: ";
```

```
    cin>>type;
```

```
    switch (type)
```

```
    {
```

```
        case 1:
```

```
            cout<<"Enter principal: ";
```

```
            cin>>p;
```

```
            cout<<"Enter rate of interest: ";
```

```
            cin>>r;
```

```
            cout<<"Enter time period in years: ";
```

```
            cin>>n;
```

```
            a=p*(1+(r/100))*n;
```

```
            cout<<"Maturity amount = "<<a;
```

```

        break;
    case 2:
        cout<<"Enter monthly installment: ";
        cin>>p;
        cout<<"Enter rate of interest: ";
        cin>>r;
        cout<<"Enter time period in months: ";
        cin>>n;
        a=(p*n)+(p*n*(n+1)/2*r/100*1/12);
        cout<<"Maturity amount = "<<a;
        break;
    default:
        cout<<"Invalid Choice.\n";
        break;
}
return 0;
}

```

\*\*\*\*\*OUTPUT-1\*\*\*\*\*

```

1. Term Deposit
2. Recurring Deposit
Enter your choice: 1
Enter principal: 1000
Enter rate of interest: 5
Enter time period in years: 2
Maturity amount = 2100

```

\*\*\*\*\*OUTPUT-2\*\*\*\*\*

```

1. Term Deposit
2. Recurring Deposit
Enter your choice: 2
Enter monthly installment: 1000

```

Enter rate of interest: 5

Enter time period in months: 2

Maturity amount = 2012.5

\*\*\*\*\*OUTPUT-3\*\*\*\*\*

1. Term Deposit

2. Recurring Deposit

Enter your choice: 3

Invalid Choice.

/\*

Name : Deepanshu Gupta

Section : AI & ML

Roll Number : 10

3. A special two-digit number is such that when the sum of its digits is added to the product of its digits, the result is equal to the original two-digit number.

Example: Consider the number 59.

Sum of digits =  $5 + 9 = 14$

Product of its digits =  $5 \times 9 = 45$

Sum of the sum of digits and product of digits =  $14 + 45 = 59$

Write a C++ program to accept a two-digit number. Add the sum of its digits to the product of its digits. If the value is equal to the number input, output the message "Special 2-digit number" otherwise, output the message "Not a Special 2-digit number".

\*/

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int num, n, s=0, p=1, res;
```

```
    cout<<"Enter a two digit number: ";
```

```
    cin>>num;
```

```
    if (num>9 && num<100)
```

```
    {
```

```
        n=num;
```

```
        while(n!=0)
```

```
        {
```

```
            s=s+(n%10);
```

```
            p=p*(n%10);
```

```
            n=n/10;
```

```
        }
```

```
        cout<<"Sum of dgits: "<<s;
```

```
        cout<<"\nProduct of digits: "<<p;
```

```
        res=s+p;
```

```
    cout<<"\nSum of the sum of digits and product of digits: "<<res;
    if (num==res)
        cout<<"\nSpecial 2-digit number";
    else
        cout<<"\nNot a special 2-digit number";
}
else
    cout<<"Entered number is not a 2-digit number";
return 0;
}
```

\*\*\*\*\*OUTPUT\*\*\*\*\*

Enter a two digit number: 59

Sum of dgits: 14

Product of digits: 45

Sum of the sum of digits and product of digits: 59

Special 2-digit number



/\*

Name : Deepanshu Gupta

Section : AI & ML

Roll Number : 10

4. WAP in C++ to find the smallest digit of an integer that is input:

Sample input: 6524

Sample output: smallest digit is 2

\*/

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int num, min;
```

```
    cout<<"Enter a number: ";
```

```
    cin>>num;
```

```
    min=num%10;
```

```
    num/=10;
```

```
    while (num!=0)
```

```
    {
```

```
        if (min>(num%10))
```

```
            min=num%10;
```

```
            num/=10;
```

```
    }
```

```
    cout<<"Smallest digit: "<<min;
```

```
    return 0;
```

```
}
```

\*\*\*\*\*OUTPUT\*\*\*\*\*

Enter a number: 3124

Smallest digit: 1

/\*

Name : Deepanshu Gupta

Section : AI & ML

Roll Number : 10

5. An Evil number is appositive whole number which has even number of 1's in its binary equivalent.

Example: Binary equivalent of 9 is 1001, which contains even number of 1's.

Thus, 9 is evil number.

A few evil numbers are 3, 5, 6, 9....

Design a program to accept a positive whole number 'N' where  $N > 2$  and  $N < 100$ . Find the binary equivalent of the number and count the number of 1's in it and display whether it is an Evil number or not with an appropriate message.

\*/

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int num, n, b=0, i=1, count=0;
```

```
    cout<<"Enter decimal number: ";
```

```
    cin>>num;
```

```
    n=num;
```

```
    while (n>0)
```

```
    {
```

```
        b+=i*(n%2);
```

```
        n/=2;
```

```
        i*=10;
```

```
    }
```

```
    cout<<"Binary form: "<<b;
```

```
    n=b;
```

```
    while (n>0)
```

```
    {
```

```
        if(n%10==1)
```

```
            count++;
```

```
        n/=10;
    }
    if (count%2==0)
        cout<<"\nNumber is evil";
    else
        cout<<"\nNumber is not evil";
    return 0;
}
```

\*\*\*\*\*OUTPUT-1\*\*\*\*\*

Enter decimal number: 11

Binary form: 1011

Number is not evil

\*\*\*\*\*OUTPUT-2\*\*\*\*\*

Enter decimal number: 12

Binary form: 1100

Number is evil