Name: Patel Manan Maheshkumar
Roll No: CE-111
PPS-II(Lab-4)

1) Caeser Cipher

```
#include<iostream>
using namespace std;
int main()
{
   int n,key;
   cin >> n >> key;
   string s;
   for(int i=0;i<n;i++){
      cin >> s;
      for(int j=0;s[j]!='\0';j++)
      {
         if(s[j]>='0' && s[j]<='9')
         s[j]=(s[j]-'0'+key)%10 + '0';
         if(s[j]>='A' && s[j]<='Z')
         s[j]=(s[j]-'A'+key)%26 + 'A';
         if(s[j]>='a' && s[j]<='z')
         s[j]=(s[j]-'a'+key)%26 + 'a';
      }
   cout << s << endl;
}
return 0;
}
```

2) Concatenate all the strings in the input with space in between

```
#include <string>
#include <iostream>
using namespace std;
int main()
{
   string s,finals;
   int flag=1;
   while(getline(cin>>ws, s))
   {
      if(tolower(s[0]))
         s[0]=toupper(s[0]);
      if(flag)
      {
         finals=s;
         flag=0;
```

```
      }
      else
         finals+=(" "+s);
   }
   cout << finals;
   return 0;
}
```

3) Playing with C++ string object

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
   string str;
   string substr;
   int n,count=0;

   getline(cin >> ws,str);
   cin >> n;
   getline(cin >> ws,substr);

   int l=substr.size();

   for(int i=1;i<n;i++)
      str+=str;
   while(true)
   {
      int index = str.find(substr);
      if(index==-1) break;
         count++;
      str.erase(index,l);
   }
   cout << count << endl;
return 0;
}
```

4) Playing with C++ string object 2

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
   string str;
   string substr;
```

```
   int n;

   getline(cin >> ws,str);
   cin >> n;
   getline(cin >> ws,substr);

   for(int i=1;i<n;i++)
      str+=str;
   int index=str.find(substr);

   while(index!=-1)
   {
      if(index!=-1)
         cout << index << " ";
   index=str.find(substr,index+1);
   }
return 0;
}
```

5) Moving point on 2-D plane

```
#include <iostream>
class MovingPoint
{
   int x;
   int y;
   int flag=1;
   public:
   void initialize(int x, int y)
   {
      if(flag)
      {
         this->x=x;
         this->y=y;
         flag=0;
      }
   }
   void move_left(int l)
   {
      x-=l;
   }
   void move_right(int r)
   {
      x+=r;
   }
   void move_up(int u)
   {
```

```
    y+=u;
  }
  void move_down(int d)
  {
    y-=d;
  }
  void print_current_position()
  {
    std::cout << x << " " << y;
  }
};
int main()
{
 MovingPoint mp;
 int x, y, n, i, units;
 char direction;

 std::cin >> x >> y;
 mp.initialize(x, y);

 std::cin >> n;
 for(i = 0; i < n; i++)
 {
  std::cin >> direction >> units;
  switch(direction)
  {
   case 'L':
    mp.move_left(units);
    break;
   case 'R':
    mp.move_right(units);
    break;
   case 'U':
    mp.move_up(units);
    break;
   case 'D':
    mp.move_down(units);
    break;
  }
 }
 mp.initialize(0, 0);
 mp.print_current_position();

 return 0;
}
```

6) Bonus_Calculator

```
#include <iostream>
#include <cmath>
using namespace std;
int e_total=0,m_total=0,t_total=0,total=0;
class Engineer
{
   int salary;
   string name;
   public:
   void initialize(int salary, string s)
   {
      this->salary=salary;
      this->name=s;
      e_total+=salary;
   }
   void print()
   {
      cout << " " << name << " " << salary << endl;
   }
   void bonus()
   {
      e_total*=0.25;
      total+=e_total;
      cout << e_total << " ";
   }
};
class Manager
{
   int salary;
   string name;
   public:
   void initialize(int salary, string s)
   {
      this->salary=salary;
      this->name=s;
      m_total+=salary;
   }
   void print()
   {
      cout << " " << name << " " << salary << endl;
   }
   void bonus()
   {
      m_total*=0.3;
      total+=m_total;
      cout << m_total << " ";
```

```
    }
};
class TeamLeader
{
    int salary;
    string name;
    public:
    void initialize(int salary, string s)
    {
        this->salary=salary;
        this->name=s;
        t_total+=salary;
    }
    void print()
    {
        cout << " " << name << " " << salary << endl;
    }
    void bonus()
    {
        t_total*=0.28;
        total+=t_total;
        cout << t_total << " ";
    }
};
int main()
{
    int n;
    cin >> n;
    int x=0;
    for(int i=0;i<n;i++)
    {
        x++;
        char emp;
        cin >> emp;
        int salary=0;
        string s1;
        Engineer e;
        Manager m;
        TeamLeader t;
        if(emp=='E')
        {
            cin >> salary;
            getline(cin>>ws,s1);
            e.initialize(salary,s1);
            e.print();
        }
        if(emp=='M')
```

```
      {
         cin >> salary;
         getline(cin>>ws,s1);
         m.initialize(salary,s1);
         m.print();
      }
      if(emp=='T')
      {
         cin >> salary;
         getline(cin>>ws,s1);
         t.initialize(salary,s1);
         t.print();
      }
      if(x==n)
      {
         e.bonus();
         t.bonus();
         m.bonus();
         cout << total;
      }
   }
return 0;
}
```