SOFTWARE PROJECT MANAGEMENT LAB - G2
EXPERIMENT 9

- ASHISH KUMAR
- 2K18/SE/041

AIM:- Given the actors and use cases present in a software system, implement the program to compute the use case points of the project.

THEORY:-

A Use-Case is a series of related interactions between a user and a system that enables the user to achieve a goal.

Use-Cases are a way to capture functional requirements of a system. The user of the system is referred to as an ‘Actor’. Use-Cases are fundamentally in text form.

Use-Case Points (UCP) is a software estimation technique used to measure the software size with use cases. The concept of UCP is similar to FPs.

The Use-Case Points counting process has the following steps −

- Calculate unadjusted UCPs
- Adjust for technical complexity
- Adjust for environmental complexity
- Calculate adjusted UCPs

Unadjusted Use-Case Points (UUCP) = UUCW + UAW

Impact of the Factor = Impact Weight × Rated Value

TCF = 0.6 + (0.01 × TFactor)

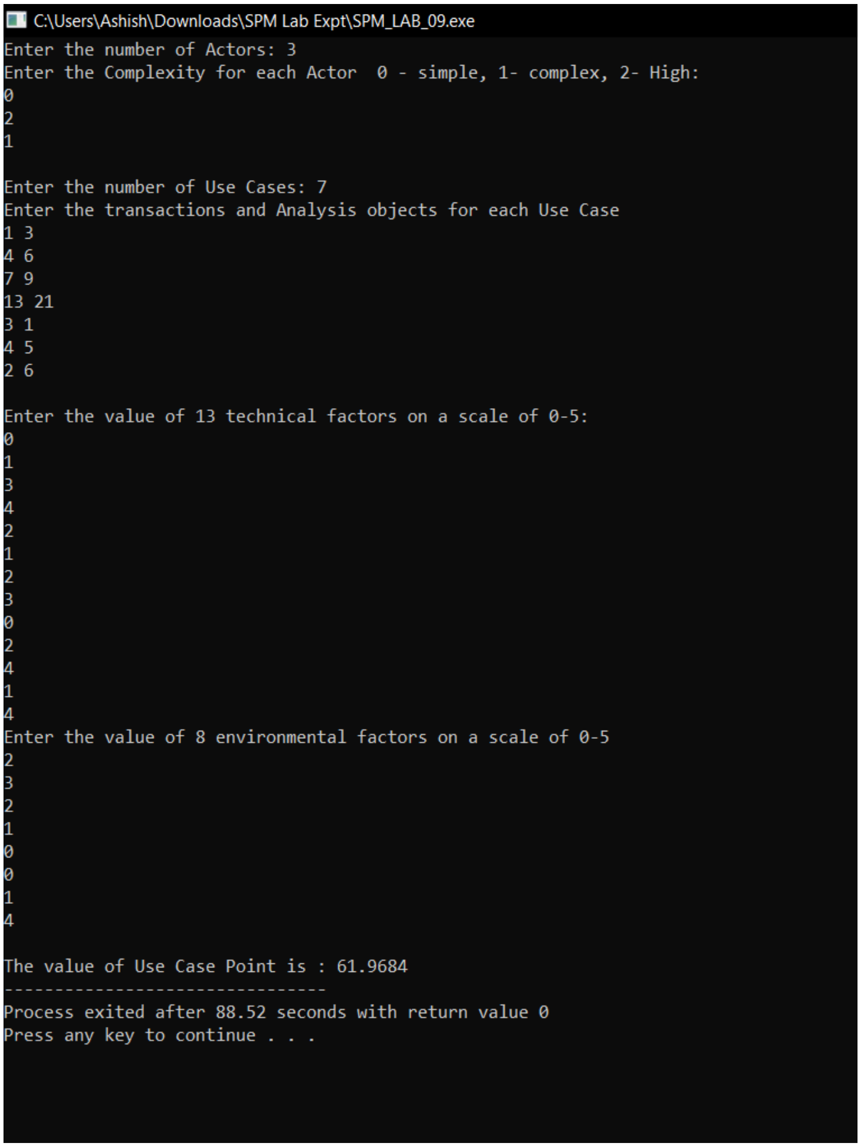
Environmental Factor (EF) = 1.4 + (-0.03 × EFactor)

Adjusted Use-Case Points, UCP = UUCP × TCF × EF

CODE:-

```
#include<iostream>
#include<bits/stdc++.h>
#define ll long long
using namespace std;

int main() {

  //Actors
  int Ano;
  cout << "Enter the number of Actors: ";
  cin >> Ano;
  int actors[Ano];
  cout << "Enter the Complexity for each Actor  0 - simple, 1- complex, 2- High:\n";
  for (int i = 0; i < Ano; i++)
    cin >> actors[i];

  //UAW
  float UAW = 0;
  for (int i = 0; i < Ano; i++)
    UAW += actors[i] + 1;

  //Use cases
  int UCno;
  cout << "\nEnter the number of Use Cases: ";
  cin >> UCno;
  int use_cases[UCno][2];
  cout << "Enter the transactions and Analysis objects for each Use Case \n";
  for (int i = 0; i < UCno; i++)
    cin >> use_cases[i][0] >> use_cases[i][1];
  int use_cases_weight_factor[UCno];
  for (int i = 0; i < UCno; i++) {
    if (use_cases[i][0] <= 3 || use_cases[i][1] <= 5) {
      use_cases_weight_factor[i] = 5;
    }
    else if(use_cases[i][0] < 7 || use_cases[i][1] < 10) {
      use_cases_weight_factor[i] = 10;
    }
    else {
      use_cases_weight_factor[i] = 15;
    }
  }
  //UUCW
  float UUCW = 0;
  for (int i = 0; i < UCno; i++)
    UUCW += use_cases_weight_factor[i];

  //UUCP
  float UUCP = UAW + UUCW;

  //FCF
  cout << "\nEnter the value of 13 technical factors on a scale of 0-5: \n";
  float Wi[13] = {2,1,1,1,1,0.5,0.5,2,1,1,1,1,1};
  float Fi[13];
  float TCF = 0;
  for (int i = 0; i < 13; i++)
    cin >> Fi[i];
  for (int i = 0; i < 13; i++)
    TCF += Fi[i] * Wi[i];
  TCF *= 0.01;
  TCF += 0.6;

  //ECF
  cout << "Enter the value of 8 environmental factors on a scale of 0-5\n";
  float Wj[13] = {1.5, 0.5, 0.5, 1, 1, -1, -1, 2 };
  float Fj[13];
  float ECF = 0;
  for (int i = 0; i < 8; i++)
    cin >> Fj[i];
  for (int i = 0; i < 8; i++)
    TCF += Fj[i] * Wj[i];
  TCF *= -0.03;
  TCF += 1.4;

  //UCP
  float UCP = UUCP + TCF + ECF;
  cout << "\nThe value of Use Case Point is : " << UCP;
  return 0;
}
```

OUTPUT:-



```
Enter the number of Actors: 3
Enter the Complexity for each Actor  0 - simple, 1- complex, 2- High:
0
2
1

Enter the number of Use Cases: 7
Enter the transactions and Analysis objects for each Use Case
1 3
4 6
7 9
13 21
3 1
4 5
2 6

Enter the value of 13 technical factors on a scale of 0-5:
0
1
3
4
2
1
2
3
0
2
4
1
4
Enter the value of 8 environmental factors on a scale of 0-5
2
3
2
1
0
0
1
4

The value of Use Case Point is : 61.9684
--------------------------------
Process exited after 88.52 seconds with return value 0
Press any key to continue . . .
```

Findings & Learning: -

- We have successfully implemented Use Case Point.
- And we learned following things:
    i. Environmental factors do affect the effort that would be required to develop software.
    ii. Actors affect the use case point value depending upon the weights assigned to them.
    iii. Use Case Point is one of the techniques with largest amount of factor coverage.