LAB REPORT

CSE 114 : Data Structure and Algorithms Sessional

PREPARED BY

Mehrin Farzana
ID: 2101013
Session: 2021-2022
Date: 09/08/2023

SUPERVISED BY

Suman Saha
Lecturer
Department of IRE, BDU

BANGABANDHU SHEIKH MUJIBUR RAHMAN DIGITAL UNIVERSITY

(BDU)

List of Problems

1. Given the following-
• A knapsack (kind of shoulder bag) with limited weight capacity.
• Few items each having some weight and value.

The problem states,
Which items should be placed into the knapsack such that-
• The value or profit obtained by putting the items into the knapsack is maximum.
• And the weight limit of the knapsack does not exceed.
• Either take the item or leave it whole.

2. Given two string, find the longest sub sequence.

Problem No.: 01

Problem Statement:

Given the following-
• A knapsack (kind of shoulder bag) with limited weight capacity.
• Few items each having some weight and value.

The problem states,
Which items should be placed into the knapsack such that-
• The value or profit obtained by putting the items into the knapsack is maximum.
• And the weight limit of the knapsack does not exceed.
• Either take the item or leave it whole.

Code:

```
#include <stdio.h>
int max(int a, int b){
   if(a>=b)
      return a;
   else
      return b;
}

int main()
{
   printf("Enter number of items & knapsack capacity: ");
   int n,C,k=0;
   scanf("%d%d", &n, &C);

   int w[n], v[n], o[n];
   int p[n+1][C+1];

   printf("Enter weight: ");
   for(int i=0; i<n; i++)
      scanf("%d", &w[i]);

   printf("Enter value: ");
   for(int i=0; i<n; i++)
      scanf("%d", &v[i]);

   for(int i=0; i<=n; i++){
      for(int j=0; j<=C; j++){
         if(i==0 || j==0){
            p[i][j]=0;
         }
```

```
            else if(j-w[i-1]>=0){
                p[i][j]=max(p[i-1][j], v[i-1]+p[i-1][j-w[i-1]]);
            }
            else if(j-w[i-1]<0){
                p[i][j]=max(p[i-1][j],p[i][j-1]);
            }

        }

    }
    printf("Max profit: %d\n", p[n][C]);

    for(int i=n, j=C; i>0; i--){
        if(p[i][j]!=p[i-1][j]){
            o[k]=i;
            j-=w[i-1];
            k++;
        }
    }
    printf("Items included: ");
    for(int i=k-1; i>=0; i--)
        printf("%d ", o[i]);
    return 0;
}
```

Output:

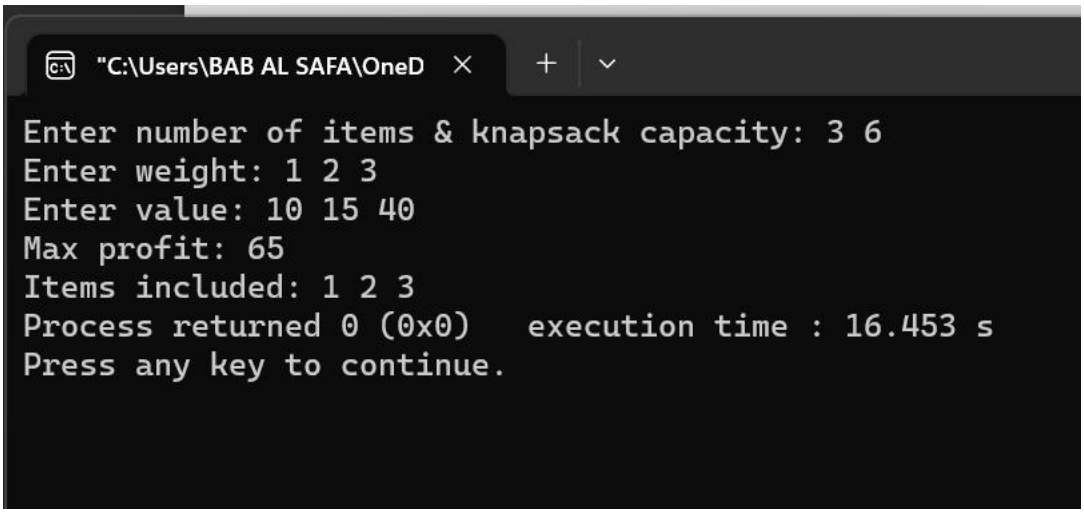

Fig 1.1: Output on console for case 1.

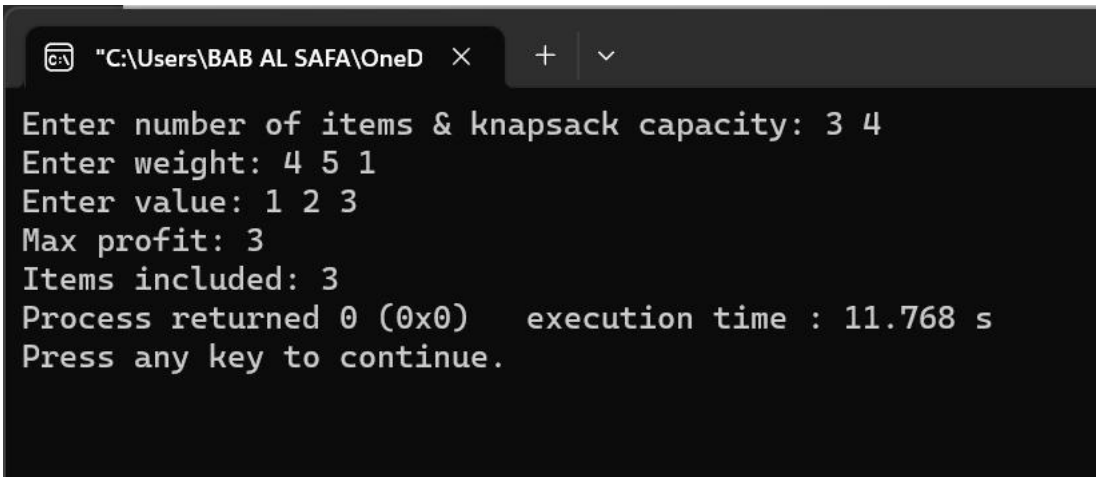

Fig 1.2: Output on console for case 2.

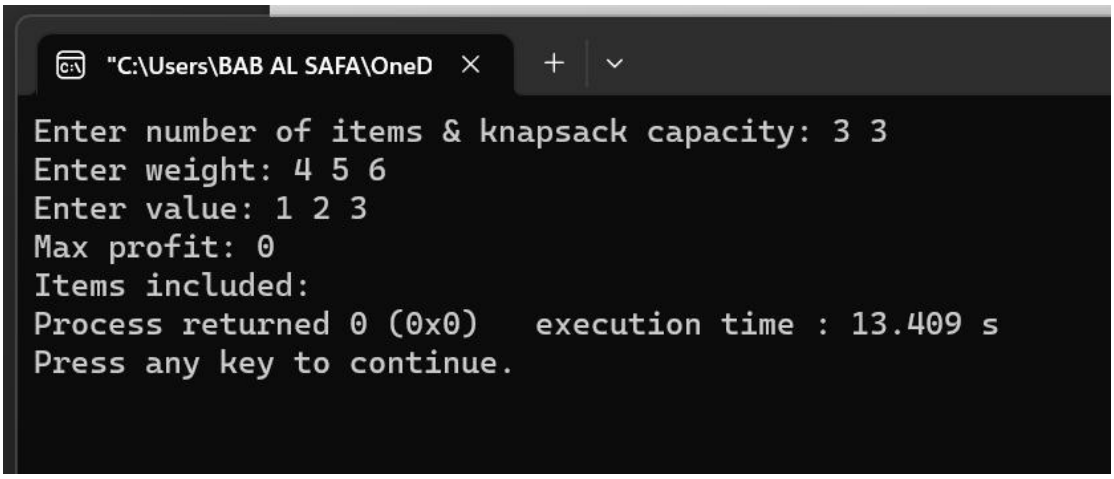

Fig 1.3: Output on console for case 3.

Problem No.: 02

Problem Statement:

Given two string, find the longest sub sequence.

Code:

```
#include <stdio.h>

int main()
{
   char t1[101], t2[101];
   scanf("%s %s", t1, t2);
   int n1=strlen(t1)+1;
   int n2=strlen(t2)+1;
   int c[n1][n2];
   char b[n1][n2];
   for(int i=0; i<n1; i++){
      for(int j=0; j<n2; j++){
         if(i==0 || j==0){
            c[i][j]=0;
         }
      }
   }
   for(int i=1; i<n1; i++){
      for(int j=1; j<n2; j++){
        if(t1[i-1]==t2[j-1]){
            c[i][j]=c[i-1][j-1]+1;
            b[i][j]='*';

         }
         else if(c[i-1][j]>=c[i][j-1]){
            c[i][j]=c[i-1][j];
            b[i][j]='^';

         }
         else{
            c[i][j]=c[i][j-1];
            b[i][j]='<';

         }
      }
   }
   printf("%d\n", c[n1-1][n2-1]);
   int n;
   n1--;
   n2--;
   if(n1>=n2)
```

```
        n=n2;
    else
        n=n1;

    char lcs[n];

    for(int i=0; n1>0 && n2>0; ){
        if(b[n1][n2]=='*'){
            lcs[i]=t1[n1-1];
            n1--;
            n2--;
            i++;

        }
        else if(b[n1][n2]=='<'){
            n2--;
        }
        else if(b[n1][n2]=='^'){
            n1--;
        }
    }
    for(int i=n-1; i>=0; i--)
        if(lcs[i]!='\0')
            printf("%c", lcs[i]);
    return 0;
}
```

Output:

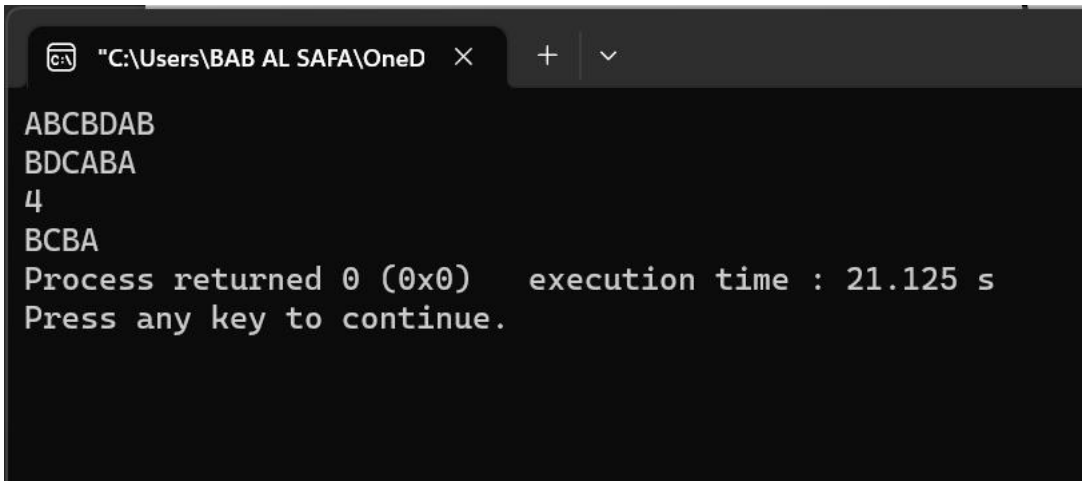

Fig 1.1: Output on console for case 1.

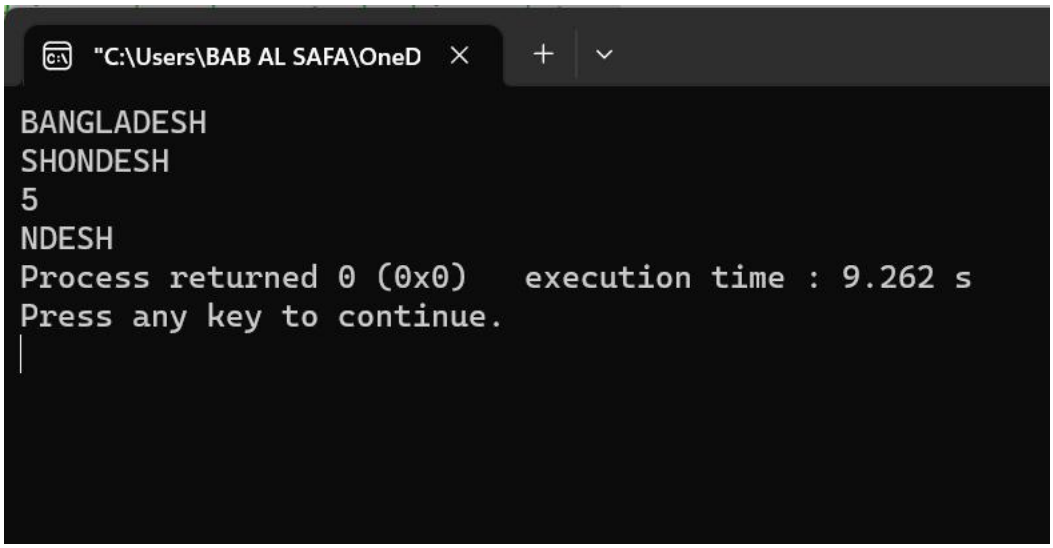

Fig 1.2: Output on console for case 2.

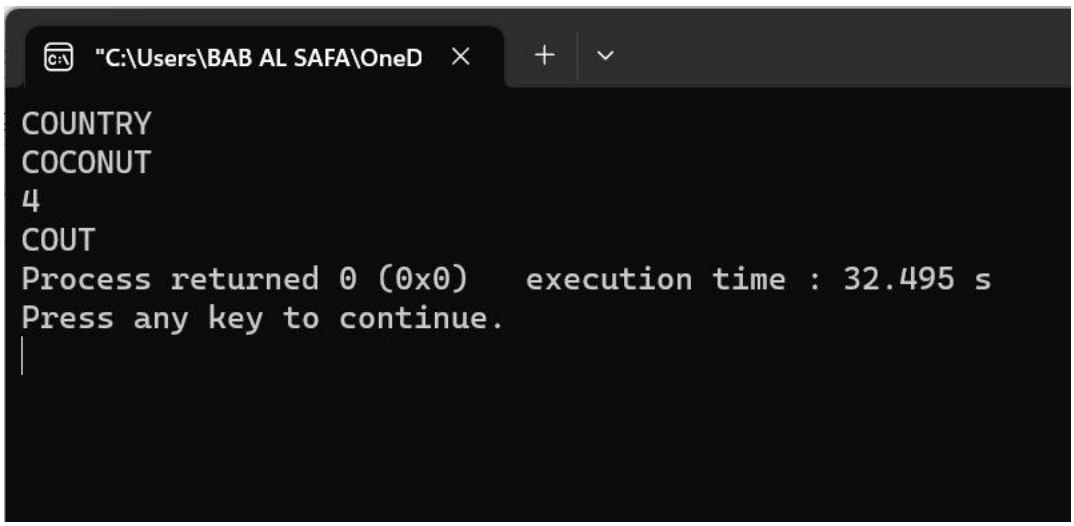

Fig 1.3: Output on console for case 3.