**0/1 KNAPSACK PROGRAM 1:**

**CODE:**

#include<stdio.h>

int max(int a, int b)

{

return (a > b)? a : b;

}

int knapSack(int W, int wt[], int val[], int n)

{

int i, w;

int K[n+1][W+1];

for (i = 0; i <= n; i++)

{

for (w = 0; w <= W; w++)

{

if (i==0 || w==0)

K[i][w] = 0;

else if (wt[i-1] <= w)

K[i][w] = max(val[i-1] + K[i-1][w-wt[i-1]], K[i-1][w]);

else

K[i][w] = K[i-1][w];

}

}

return K[n][W];

}

int main()

{

int W,n,i;

int val[10] , wt[10] ;

printf("Enter number of items:\n");

scanf("%d",&n);

printf("Enter value for each item\n");

for(i=0;i<n;i++)

scanf("%d",&val[i]);

printf("Enter weight of each item respectively\n");

for(i=0;i<n;i++)

scanf("%d",&wt[i]);

printf("Enter total weight :\n");

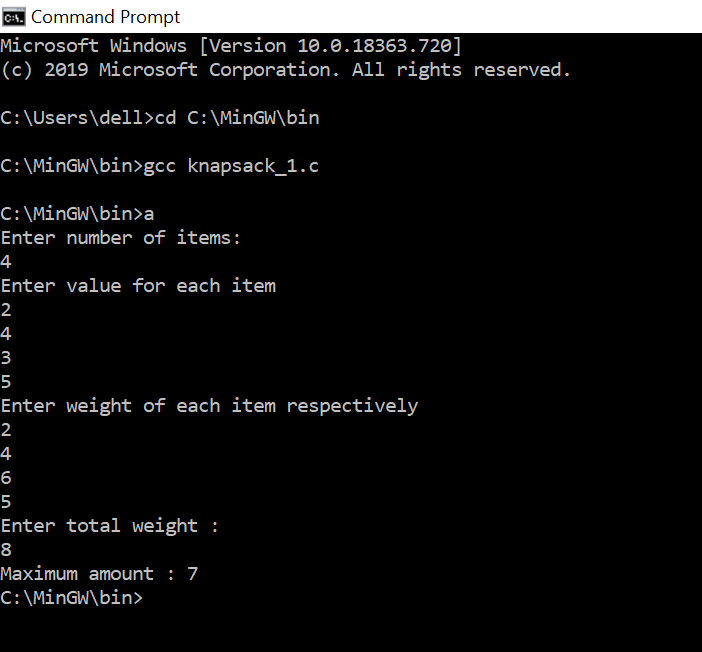
scanf("%d",&W);

printf("Maximum amount : %d", knapSack(W, wt, val, n));

return 0;

}

**OUTPUT:**



**PROGRAM 2:**

**CODE:**

#include<stdio.h>

#include<string.h>

int max (int x, int y)

{ return (x > y)? x : y; }

int longestps(char \*str)

{

int n ;

int i, j, cl;

int L[20][20];

n = strlen(str);

for (i = 0; i < n; i++)

L[i][i] = 1;

for (cl=2; cl<=n; cl++)

{

for (i=0; i<n-cl+1; i++)

{

j = i+cl-1;

if (str[i] == str[j] && cl == 2)

L[i][j] = 2;

else if (str[i] == str[j])

L[i][j] = L[i+1][j-1] + 2;

else

L[i][j] = max(L[i][j-1], L[i+1][j]);

}

}

return L[0][n-1];

}

int main()

{

char seq[20];

int n;

printf("Enter the Sequence:\n");

scanf("%s",seq);

n = strlen(seq);

printf ("The length of the Longest Palindrome Sequence is %d", longestps(seq));

getchar();

return 0;

}

**OUTPUT:**

