**CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY**

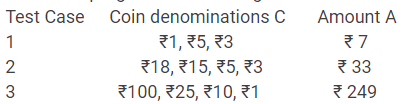
**DEVANG PATEL INSTITUTE OF ADVANCE TECHNOLOGY AND RESERACH**

**COMPUTER SCIENCE & ENGINEERING**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student ID** | 18DCS007 | **Student Name** | RUDRA BARAD |
| **Subject Code** | CE342 | **Subject Name** | Design & Analysis of Algorithms |
| **Date of exam** | 28/10/2020 |  |  |

**AIM-1:**

A cashier at any shop needs to give change of an amount to customers many times in a day. Cashier has multiple number of coins available with different denominations which is described by a set C. Implement the program for a cashier to find the minimum number of coins required to find a change of a particular amount A. Output should be the total number of coins required of given denominations. Check the program for following test cases:



Implement above problem using Greedy Approach & identify that which solution is optimal

**CODE-1:**

#include <bits/stdc++.h>

using namespace std;

void findCoins(int sum,int length,int coins[])

{

int ans[sum],j=0,i=0;

for(i=length-1;i>=0;i--)

{

while(sum>=coins[i])

{

sum=sum-coins[i];

ans[j]=coins[i];

j++;

}

}

cout<<"\nTotal coins required are : "<<j<<"\nCoins are : ";

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

cout<<ans[i]<<" ";

cout<<"\n";

}

int main()

{

int length,i,sum;

cout<<"\nEnter the different types of coins : ";

cin>>length;

int coins[length];

cout<<"Enter the denomination of coins : ";

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

cin>>coins[i];

int n = sizeof(coins)/sizeof(coins[0]);

sort(coins,coins+n);

cout<<"Enter the final sum of coins : ";

cin>>sum;

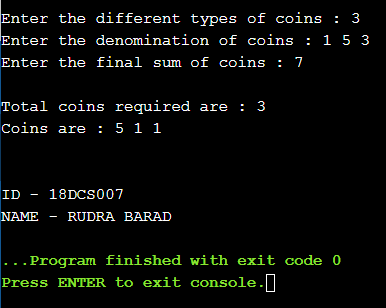
findCoins(sum,length,coins);

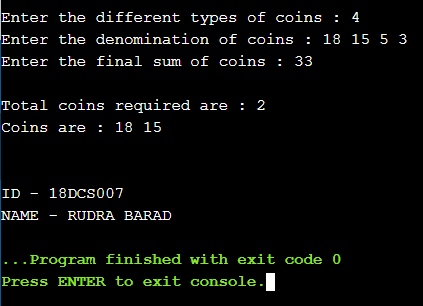
cout<<"\n\nID - 18DCS007\nNAME - RUDRA BARAD";

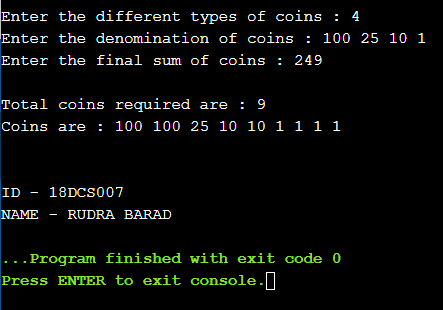
return 0;

}

**OUTPUT-1:**



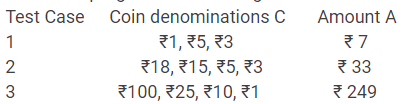




|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Denomination | Amount | No. of coins | Optimum Sol. |
| 1 | Rs 1,5,3 | Rs 7 | 3 | YES |
| 2 | Rs 18,15,3,1 | Rs 33 | 2 | YES |
| 3 | Rs 100,25,10,1 | Rs 249 | 9 | YES |

**AIM-2:**

A cashier at any shop needs to give change of an amount to customers many times in a day. Cashier has multiple number of coins available with different denominations which is described by a set C. Implement the program for a cashier to find the minimum number of coins required to find a change of a particular amount A. Output should be the total number of coins required of given denominations. Check the program for following test cases:



Implement above problem using Dynamic Approach & identify that which solution is optimal.

**CODE-2:**

#include<stdio.h>

#define MIN(a,b) a<b?a:b

void coinCount(int n, int coins[], int total)

{

int i, j, T[n+1][total+1];

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

{

for(j=0;j<=total;j++)

{

if(i==0 || j==0)

{

T[i][j] = j;

}

else if(j>=coins[i])

{

T[i][j] = MIN(T[i-1][j], 1+T[i][j-coins[i]]);

}

else

{

T[i][j] = T[i-1][j];

}

}

}

printf("\nRequired coins according to dynamic programming :");

int temp = T[n][total];

i = n;

j = total;

while(temp>0)

{

if(temp==T[i-1][j])

{

i--;

}

else

{

j -= coins[i];

printf("%d ",coins[i]);

}

temp = T[i][j];

}

printf("\nNumber of coins required : %d",T[n][total]);

}

main()

{

int n, coins[10], total, i, j;

printf("\nEnter number of coins : ");

scanf("%d",&n);

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

{

printf("Enter coin %d : ",i+1);

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

}

printf("\nEnter the total amount : ");

scanf("%d",&total);

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

{

for(j=0;j<n-1;j++)

{

if(coins[j]>coins[j+1])

{

int temp = coins[j];

coins[j] = coins[j+1];

coins[j+1] = temp;

}

}

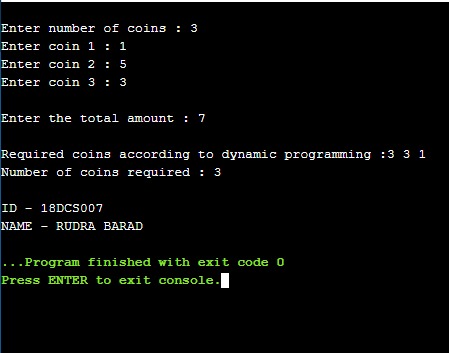
}

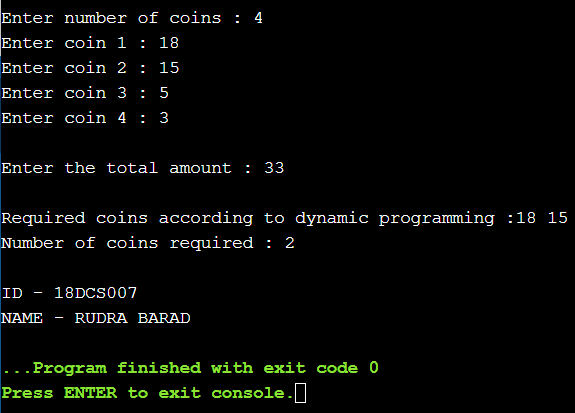
coinCount(n, coins, total);

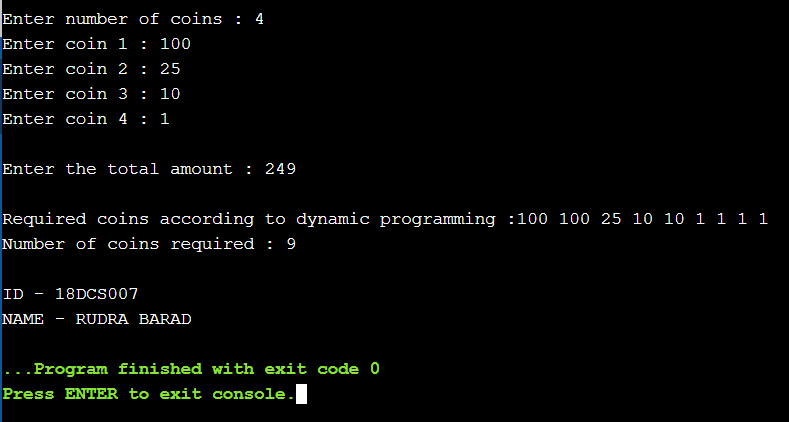
printf("\n\nID - 18DCS007\nNAME - RUDRA BARAD");

}

**OUTPUT-2:**







|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Denomination | Amount | No. of coins | Optimum Sol. |
| 1 | Rs 1,5,3 | Rs 7 | 3 | YES |
| 2 | Rs 18,15,3,1 | Rs 33 | 2 | YES |
| 3 | Rs 100,25,10,1 | Rs 249 | 9 | YES |