Ex. No: 3 **Date: 26.08.24**

Register No.: 230701346 **Name: Subasri V**

Greedy Algorithm

3.a. 1-G-Coin Problem

Aim: Write a program to take value V and we want to make change for V Rs, and we have infinite supply of each of the denominations in Indian currency, i.e., we have infinite supply of { 1, 2, 5, 10, 20, 50, 100, 500, 1000} valued coins/notes, what is the minimum number of coins and/or notes needed to make the change.

Input Format:

Take an integer from stdin.

Output Format:

print the integer which is change of the number.

Example Input :

64

Output:

4

Explanaton:

We need a 50 Rs note and a 10 Rs note and two 2 rupee coins.

Algorithm:

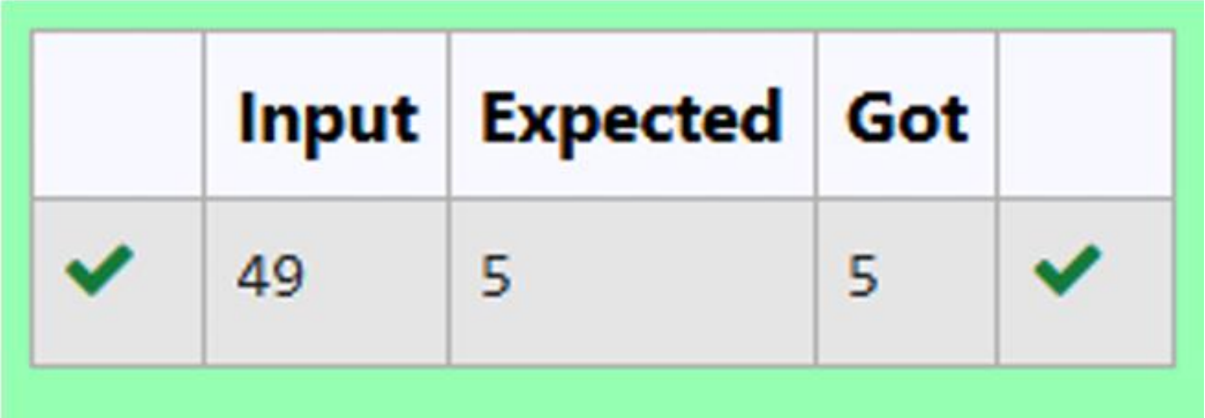
```
1. Initialize count to 0.
2. Read the integer amt from the user.
3. Initialize an array arr with the values {1, 2, 5, 10, 20, 50, 100, 500, 1000}.
4. Iterate through the array arr from the highest denomination to the lowest:
5. For each denomination arr[i]:
       a. Add the quotient of amt divided by arr[i] to count.
       b. Update amt to the remainder of amt divided by arr[i].
```

Program:

```
#include <stdio.h>
```

```
int main()
{
   int amt,count=0;
   scanf("%d",&amt);
   int arr[]={ 1, 2, 5, 10, 20, 50, 100, 500, 1000} ;
   for (int i=8;i>=0;i--)
   {
      count+=amt/arr[i];
      amt%=arr[i];
   }
   printf("%d",count);
}
```

Output:



	Input	Expected	Got	
✓	49	5	5	✓

3.b. 2-G-Cookies Problem

Aim: Assume you are an awesome parent and want to give your children some cookies. But, you should give each child at most one cookie. Each child i has a greed factor g[i], which is the minimum size of a cookie that the child will be content with; and each cookie j has a size s[j]. If s[j] >= g[i], we can assign the cookie j to the child i, and the child i will be content. Your goal is to maximize the number of your content children and output the maximum number.

Example 1:

Input:

3 1 2 3 2 1 1

Output: 1

Explanation:

You have 3 children and 2 cookies. The greed factors of 3 children are 1, 2, 3. And even though you have 2 cookies, since their size is both 1, you could only make the child whose greed factor is 1 content. You need to output 1.

Constraints:

1 <= g.length <= 3 * 10^4 0 <= s.length <= 3 * 10^4 1 <= g[i], s[j] <= 2^31 - 1

Algorithm:

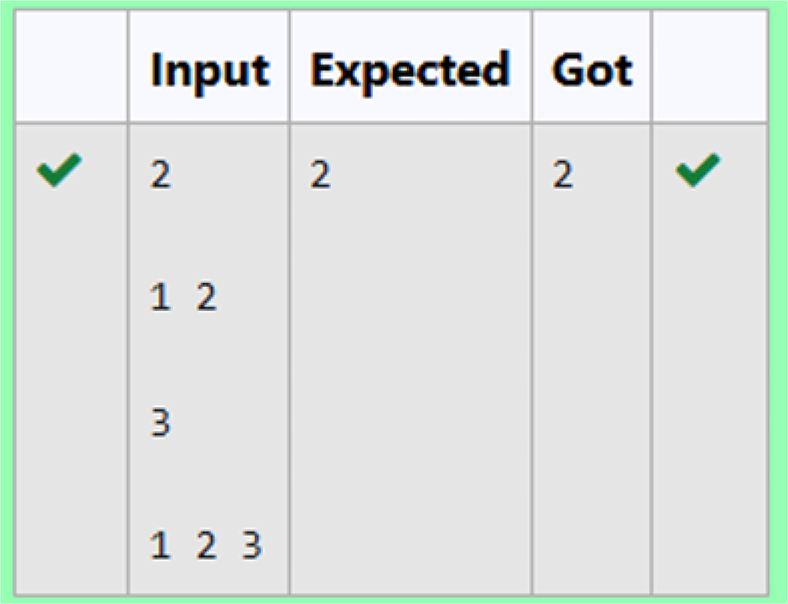
```
1. Initialize count to 0.
2. Read the integer m from the user.
3. Initialize an array g of size m to store the greed factors.
4. Read m integers into the array g.
5. Read the integer n from the user.
6. Initialize an array s of size n to store the sizes of the cookies.
7. Read n integers into the array s.
8. For each i from 0 to m-1:
9. For each j from 0 to n-1:
      a. If g[i] is less than or equal to s[i]:
            i. Increment count by 1.
            ii. Increment i by 1.
10. Print count - 1
```

Program:

```
#include <stdio.h>

int main(){
  int m,n,count=0;
  scanf("%d",&m);
  int g[m];
  for (int i=0;i<m;i++)
  {
    scanf("%d",&g[i]);
  }
  scanf("%d",&n);
  int s[n];
  for (int i=0;i<n;i++)
  {
    scanf("%d",&s[i]);
  }
  for (int i=0;i<m;i++)
  {
    for (int j=0;j<n;j++)
    {
      if (g[i]<=s[i])
      {
        count++;
        i++;
      }
    }
  }
  printf("%d",count-1);
}
```

Output:



	Input	Expected	Got	
✓	2 1 2 3 1 2 3	2	2	✓