C++

Assignment – Program

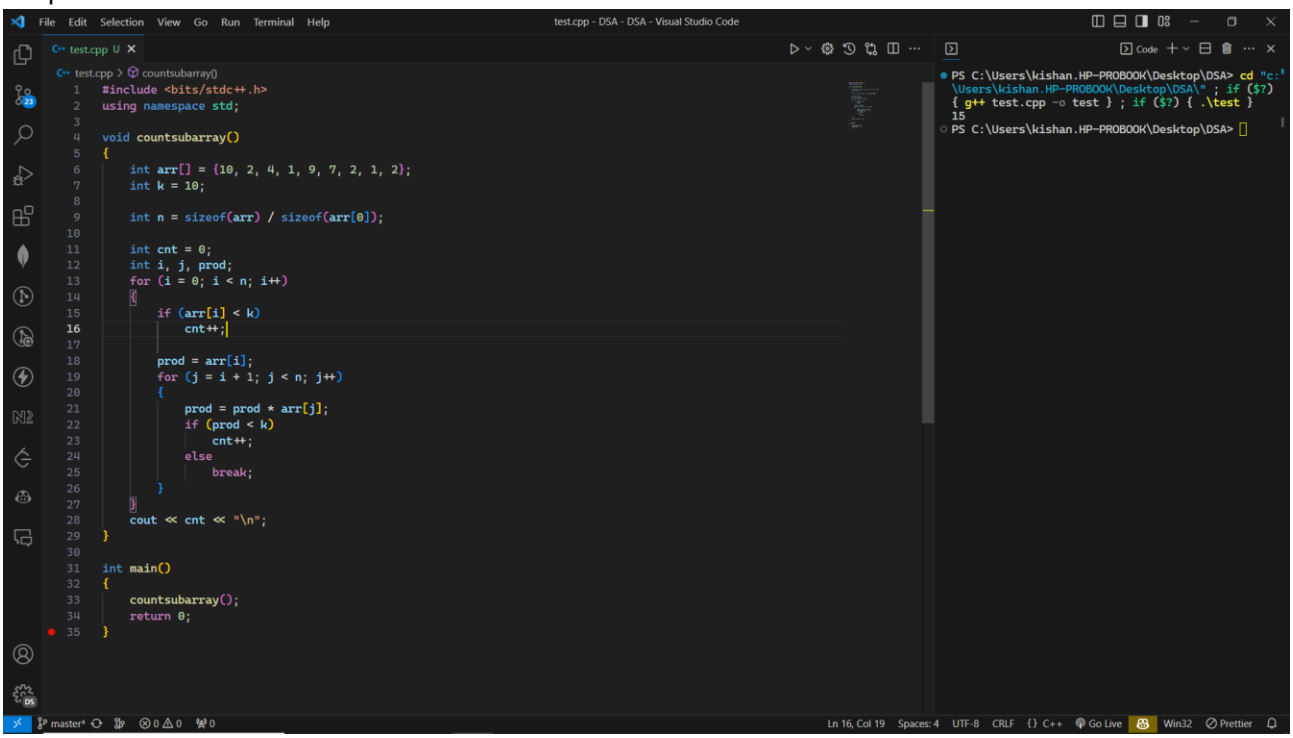
Name – Kishan R Vaghamashi

Student ID – 202312014

1)

```cpp
#include <bits/stdc++.h>
using namespace std;

void countsubarray()
{
    int arr[] = {10, 2, 4, 1, 9, 7, 2, 1, 2};
    int k = 10;

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

    int cnt = 0;
    int i, j, prod;
    for (i = 0; i < n; i++)
    {
        if (arr[i] < k)
            cnt++;

        prod = arr[i];
        for (j = i + 1; j < n; j++)
        {
            prod = prod * arr[j];
            if (prod < k)
                cnt++;
            else
                break;
        }
    }
    cout << cnt << "\n";
}

int main()
{
    countsubarray();
    return 0;
}
```

Output:

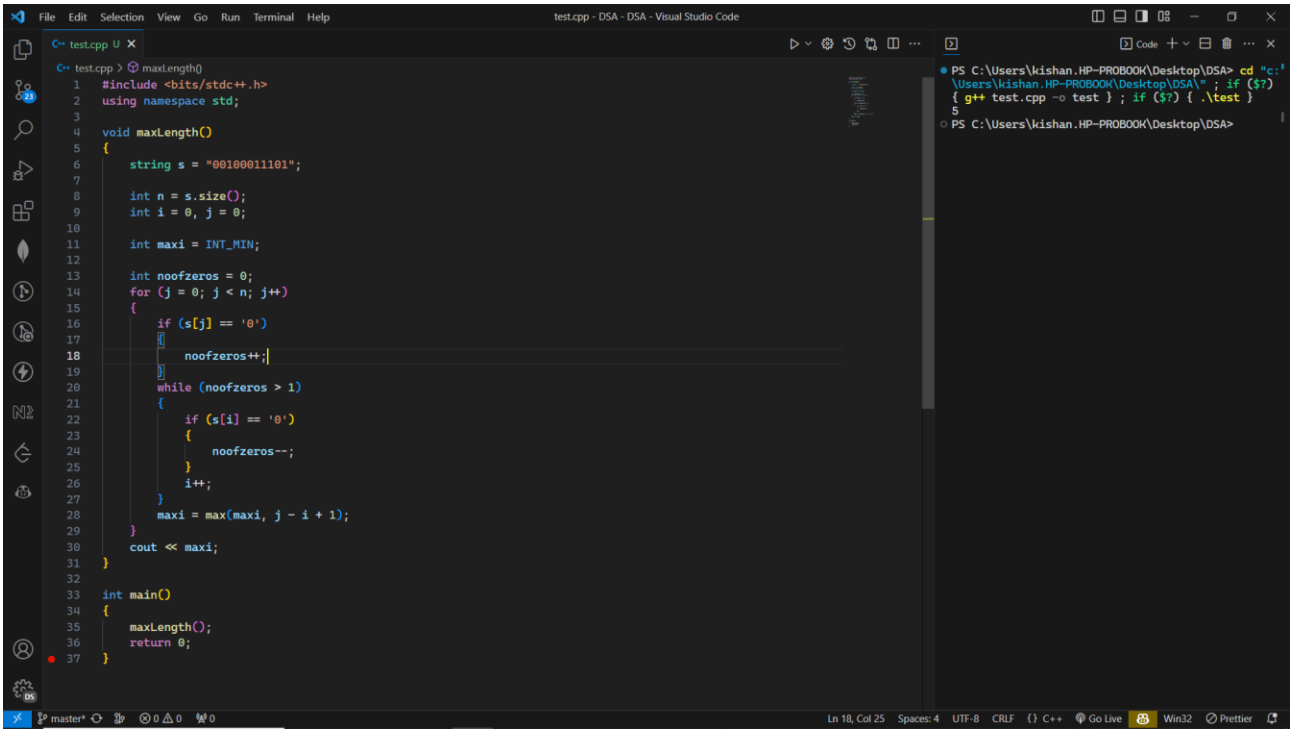


2)

```cpp
#include <bits/stdc++.h>
using namespace std;

void maxLength()
{
    string s = "00100011101";

    int n = s.size();
    int i = 0, j = 0;

    int maxi = INT_MIN;

    int noofzeros = 0;
    for (j = 0; j < n; j++)
    {
        if (s[j] == '0')
        {
            noofzeros++;
        }
        while (noofzeros > 1)
        {
```

```
            if (s[i] == '0')
            {
                noofzeros--;
            }
            i++;
        }
        maxi = max(maxi, j - i + 1);
    }
    cout << maxi;
}

int main()
{
    maxLength();
    return 0;
}
```

Output:



3)

```cpp
#include <bits/stdc++.h>
using namespace std;

void maxSubarrayLen()
{
    int arr[] = {3, 1, 2, 0, 4, 2, 1, 1, 5};
    int n = sizeof(arr) / sizeof(arr[0]);
    int target = 8;
    map<int, int> mp;

    int totalSum = 0;
    int ans = 0;
    int pos1 = -1, pos2 = -1;

    for (int i = 0; i < n; i++)
    {
        if (arr[i] == target)
        {
            ans = max(ans, 1);
        }
        totalSum += arr[i];

        int temp = totalSum - target;
        if (mp.find(temp) != mp.end())
        {
            int len = i - mp[temp];

            if (len > ans)
            {
                pos1 = mp[temp] + 1;
                pos2 = i;
                ans = len;
            }
        }
        if (mp.find(totalSum) == mp.end())
            mp[totalSum] = i;
    }
    cout << ans << "\n";
    for (int i = pos1; i <= pos2; i++)
    {
        cout << arr[i] << " ";
    }
```

```
}

int main()
{
    maxSubarrayLen();
    return 0;
}
```

Output: