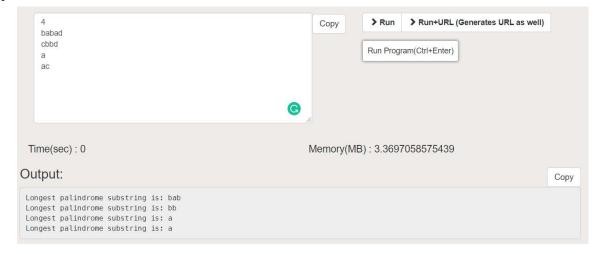
106119100 - Rajneesh Pandey

Question 1:

```
1 // 106119100 Rajneesh Pandey
 2 #include <bits/stdc++.h>
 3 using namespace std;
 5 - void printSubStr(
        string str, int low, int high)
         for (int i = low; i \le high; ++i)
            cout << str[i];}</pre>
 9 int longestPalSubstr(string str)
10 - {
11
        int n = str.size();
        bool table[n][n];
12
        memset(table, 0, sizeof(table));
13
14
        int maxLength = 1;
        for (int i = 0; i < n; ++i)
15
16
            table[i][i] = true;
17
        int start = 0;
18 -
        for (int i = 0; i < n - 1; ++i) {
19 -
            if (str[i] == str[i + 1]) {
20
                table[i][i + 1] = true;
21
                start = i;
22
                maxLength = 2;
23
            }
24
25 -
        for (int k = 3; k \le n; ++k) {
26 -
            for (int i = 0; i < n - k + 1; ++i) {
27
                int j = i + k - 1;
28 -
                if (table[i + 1][j - 1] && str[i] == str[j]) {
29
                     table[i][j] = true;
                     if (k > maxLength) {
                         start = i;
31
                         maxLength = k;
32
33
                }}}
        cout << "Longest palindrome substring is: ";</pre>
35
        printSubStr(str, start, start + maxLength - 1);
36
        return maxLength;
37 }
38 int main()
39 - { int t; cin>>t;
40 -
        while(t--){ string str; cin>>str;
41
        longestPalSubstr(str);
42
        cout<<endl;}
        return 0;}
```

Output/Input



Question 2:

```
1 // 106119100 Rajneesh Pandey
 2 #include<bits/stdc++.h>
 3 using namespace std;
 4
 5
   int dp[1001][1001];
 6
 7
    int knapSack(int W, int wt[], int val[], int n)
 8 - {
 9
       for(int i=0; i <= n; i++)
10 -
11
           for(int j=0; j<=W; j++)
            \{if(i==0 \text{ or } j==0)\}
12
13
               dp[i][j] = 0;
14
15
       for(int i=1; i <= n; i++)
16 -
17
           for(int j=1; j \leftarrow W; j++)
18 -
                 if(wt[i-1]<=j)
19
                      dp[i][j] = max(val[i-1]+dp[i-1][j-wt[i-1]], dp[i-1][j]);
20
21
22
                      dp[i][j] = dp[i-1][j];
23
24
        return dp[n][W];}
25
26 int main()
27 -
    {
28
        int t;
29
        cin>>t;
30
        while(t--)
31 -
            int n, w;
32
             cin>>n>>w;
33
             int val[n];
34
             int wt[n];
35
             for(int i=0;i< n;i++)
36
                 cin>>val[i];
37
             for(int i=0; i< n; i++)
38
                 cin>>wt[i];
39
             cout<<knapSack(w, wt, val, n)<<endl;</pre>
40
41
        return 0;
42
     }
```

Input/Output

