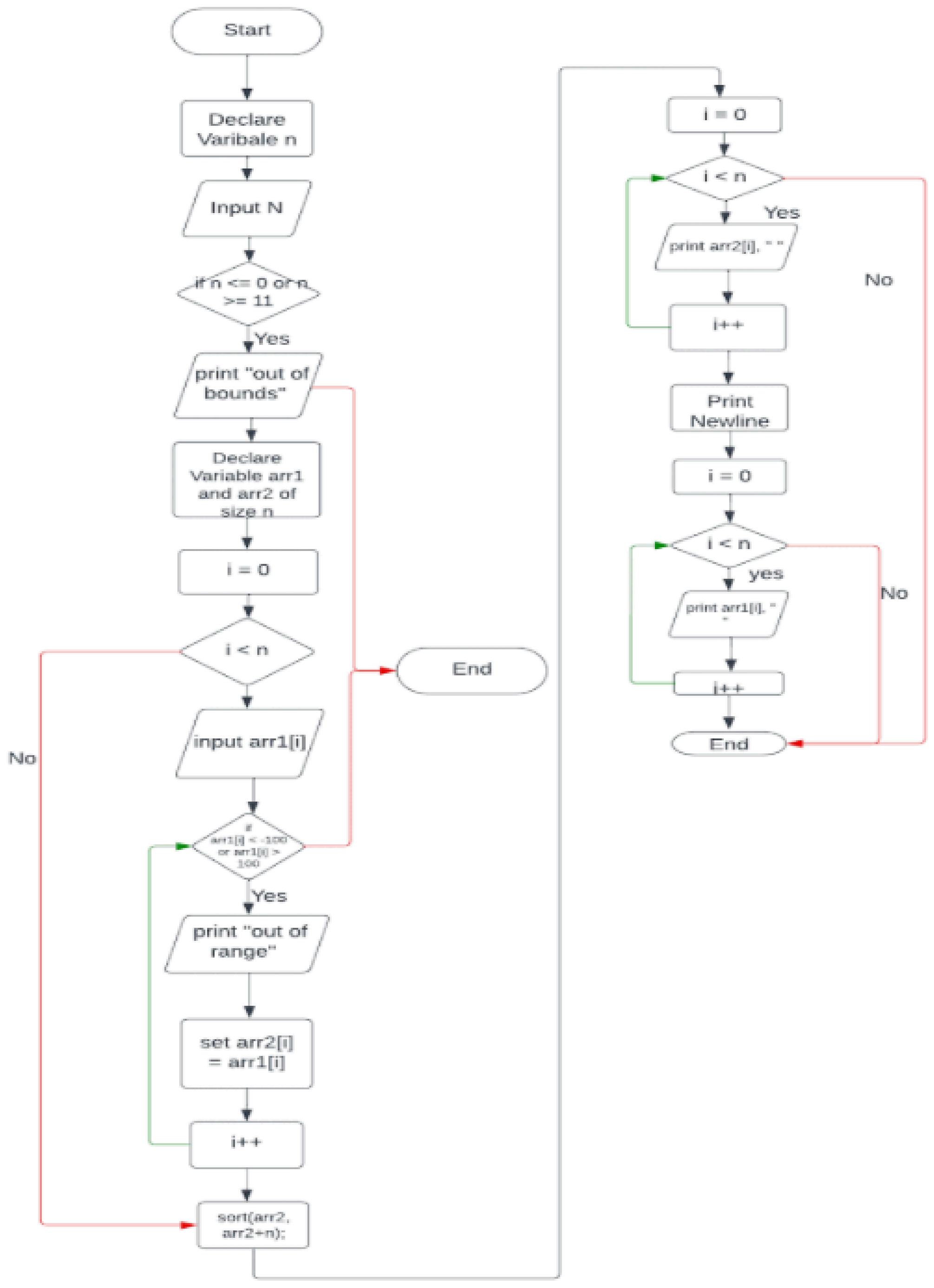
SIMPLE SORT

```
#include<iostream>
using namespace std;

int main(){
 int n;
 cin >> n;
  int a[n],b[n],sort;

  if (n == 11)
  {
   cout << "out of bounds";
   return 0;
  }
  //user input
  for( int i = 0;i < n; i++){
   cin >> a[i];
   if (a[i] > 100 || a[i] <-100){
       cout << "out of range";
   return 0;
   }
    b[i] = a[i];
  }
  //sorting
  for ( int i = 0; i < n; i++){
      for (int j = i ; j < n; j++){
          if (a[j]<a[i])
          {
           sort = a[j];
           a[j]= a [i];
           a[i] = sort;
          }
      }
  }
  //printing ordered
   for( int i = 0; i < n; i++){
      cout << a[i] << " ";
  }
     cout << endl;
      //printing un ordered
   for( int j = 0; j < n; j++){
      cout << b[j] << " ";
  }
   return 0;
}
```

Flowchart

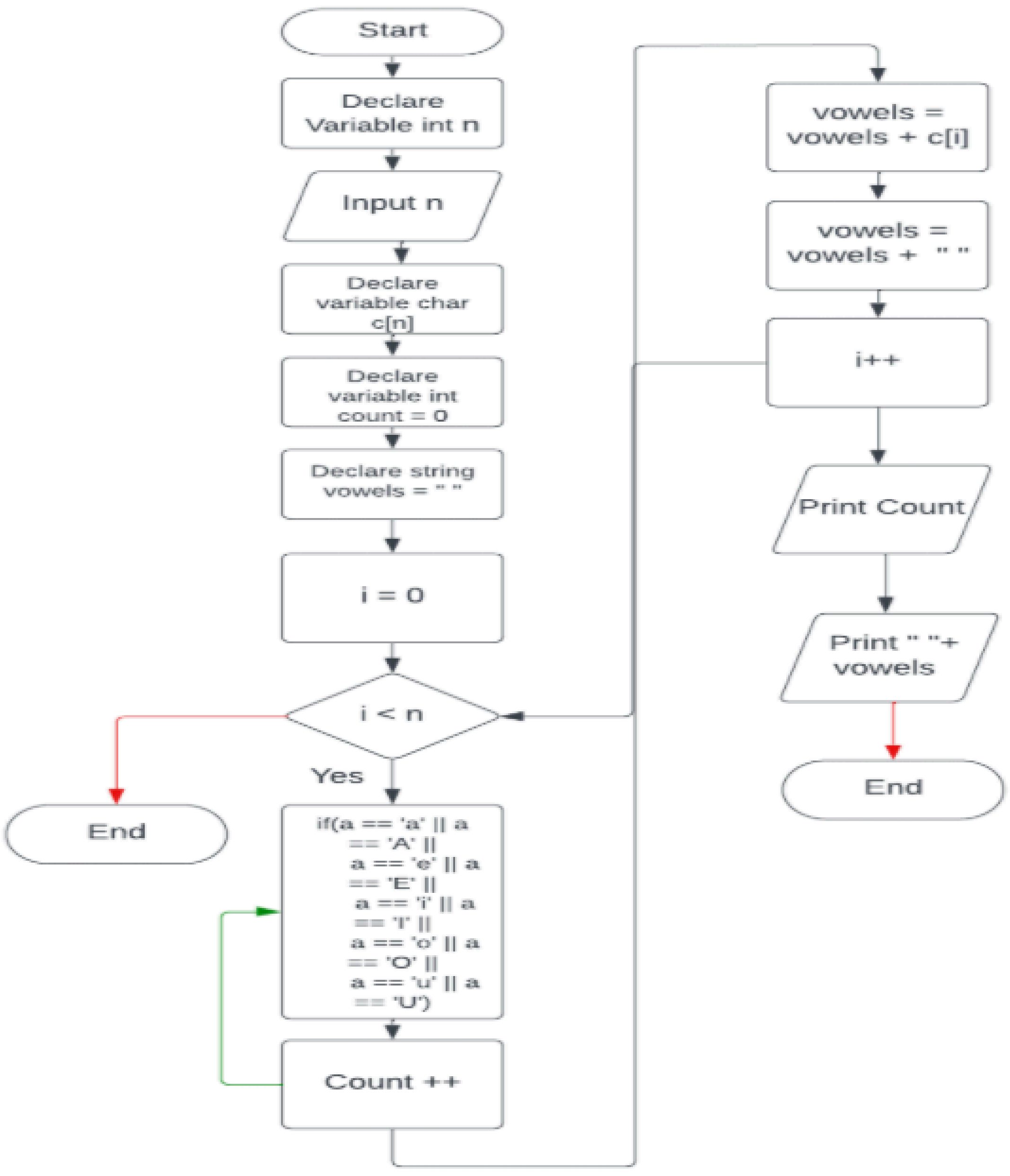


COUNTING VOWEL

```
#include <iostream>
#include <string>
using namespace std;

string vowels(int n, char c[]) {
   int count = 0;
   string v = "";
   for (int i = 0; i < n; i++) {
       if (c[i] == 'A' || c[i] == 'E' || c[i] == 'I' || c[i] == 'O' ||
c[i] == 'U' ||
           c[i] == 'a' || c[i] == 'e' || c[i] == 'i' || c[i] == 'o' ||
c[i] == 'u') {
           count++;
           v = v + c[i];
          v = v +  " ";
       }
```

```
    }
    return to_string(count)+" " + v ;
}

int main() {
    int n;
    cin >> n;
    char c[n];
    for (int i = 0; i < n; i++) {
        cin >> c[i];
    }
    cout << vowels(n, c);
    return 0;
}
```

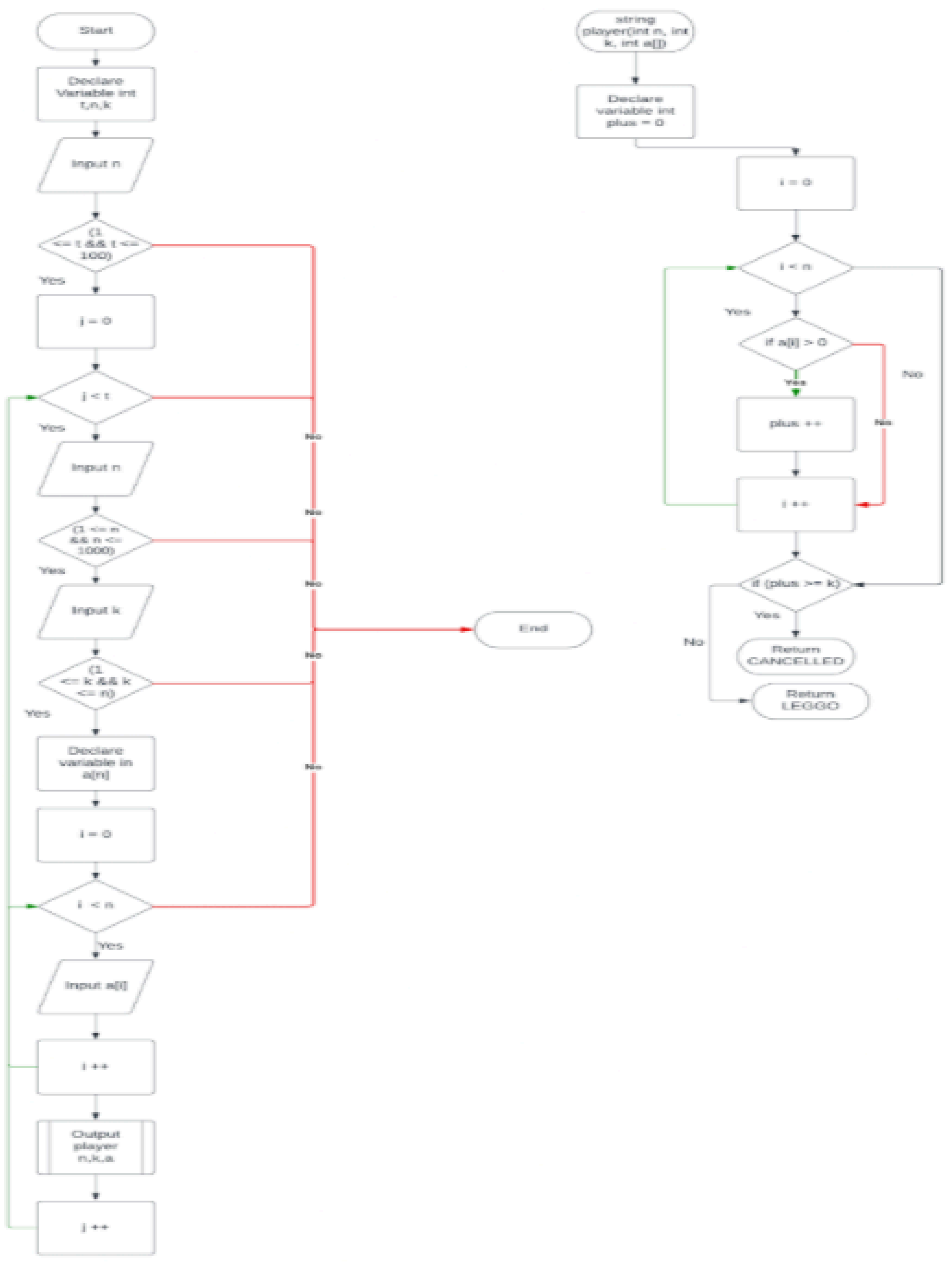
<u>Flowchart</u>


Start
Declare Variable int n
Input n
Declare variable char c[n]
Declare variable int count = 0
Declare string vowels = " "
i = 0
i < n
Yes
End
if(a == 'a' || a == 'A' || a == 'e' || a == 'E' || a == 'i' || a == 'I' || a == 'o' || a == 'O' || a == 'u' || a == 'U')
Count ++
vowels = vowels + c[i]
vowels = vowels + " "
i++
Print Count
Print " "+ vowels
End

ANGRY BIRDS

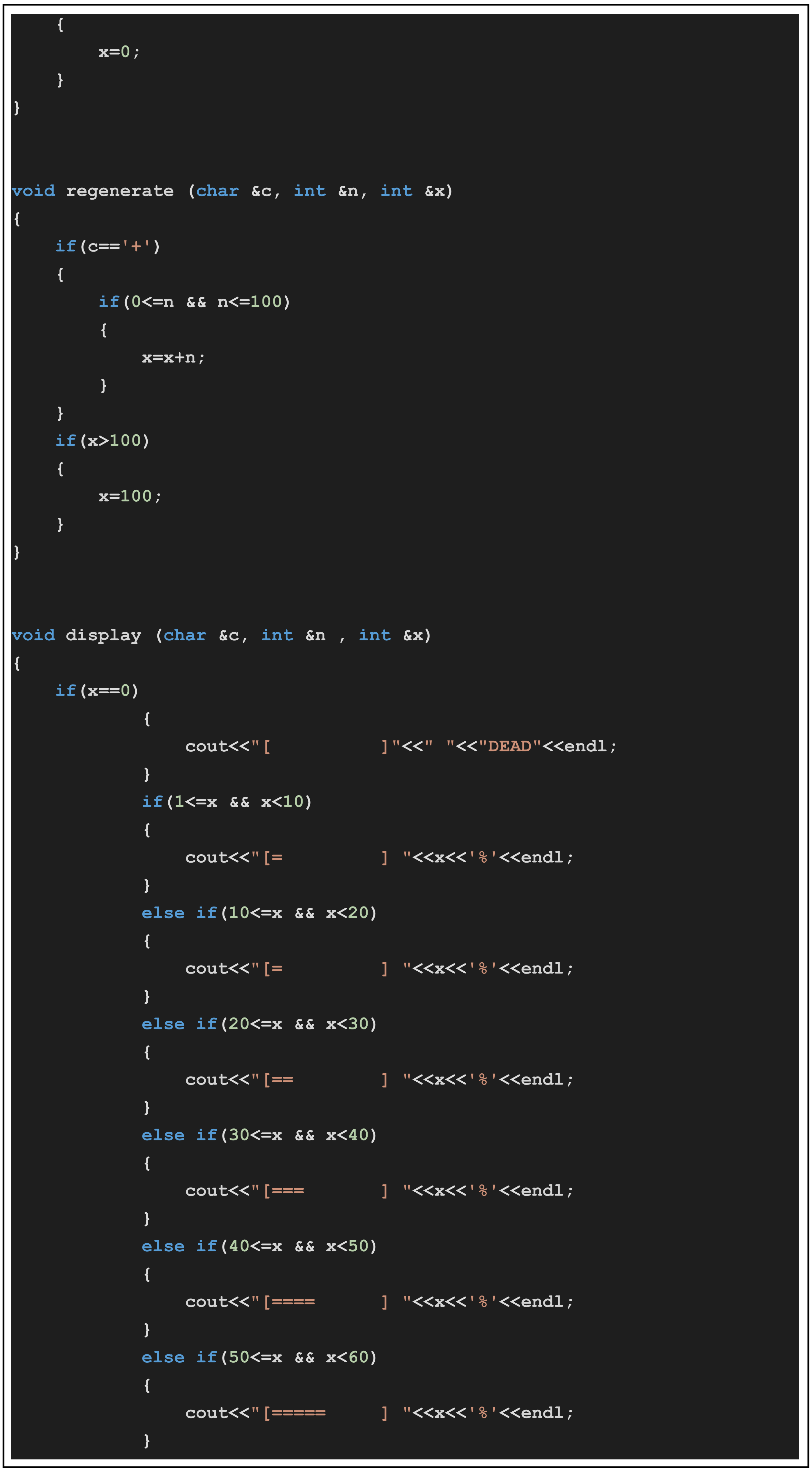
```cpp
#include <iostream>
using namespace std;

string player(int n, int k, int a[]) {
    int plus = 0;
    for (int i = 0; i < n; i++) {
        if (a[i] > 0) {
            plus++;
        }
    }
    if (plus >= k) {
        return "CANCELLED";
    }
    else {
        return "LEGGO";
    }
}

int main() {
    int t, n, k;
    cin >> t;
    for (int j = 0; j < t; j++) {
        cin >> n >> k;
        int a[n];
        for (int i = 0; i < n; i++) {
            cin >> a[i];
        }
        cout << player(n, k, a) << endl;
    }
    return 0;
}
```

Flowchart

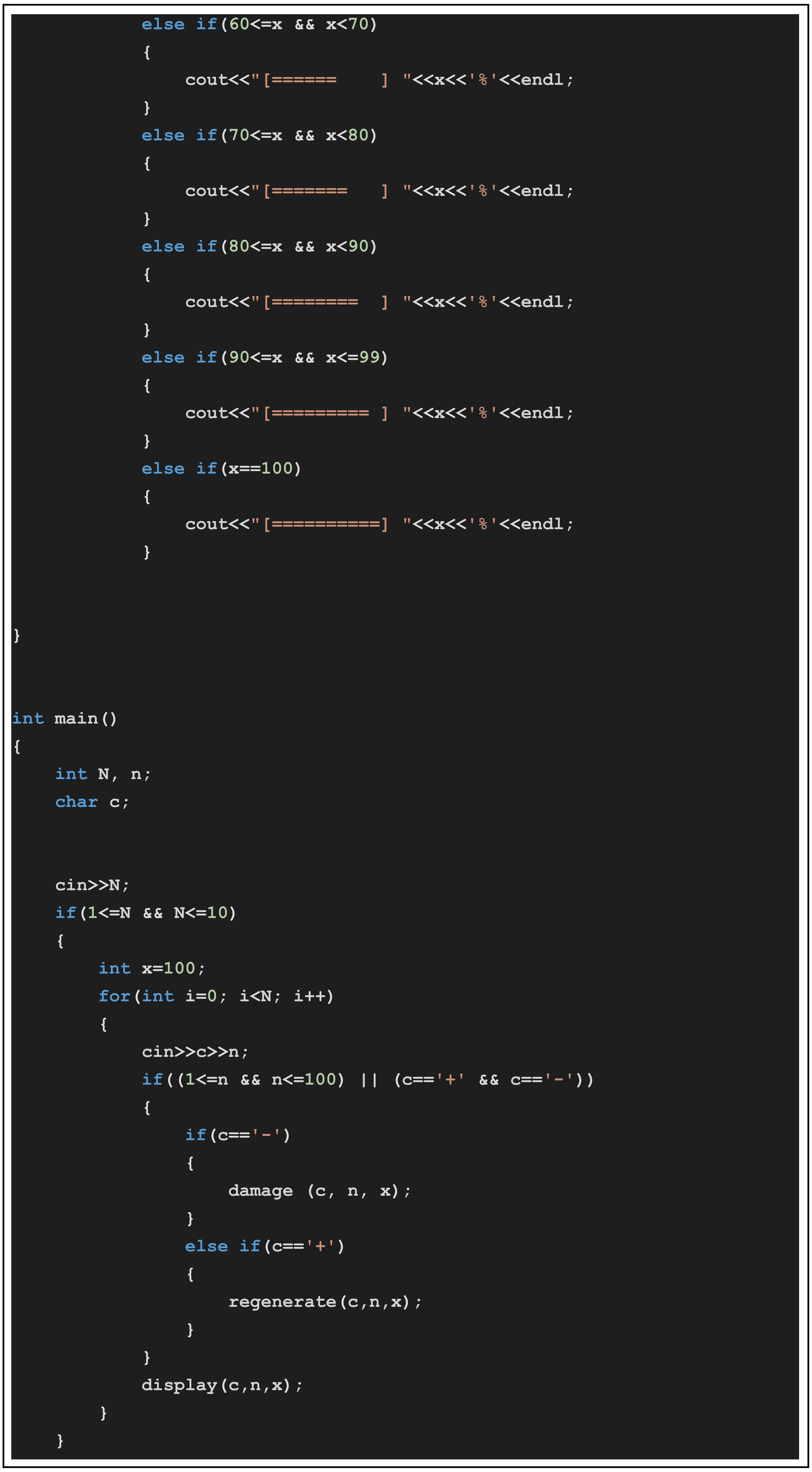


HEALTH BAR

```
#include <iostream>
using namespace std;


void damage(char &c, int &n, int &x)
{
    if(c=='-')
    {
        if(0<=n && n<=100)
        {
            x=x-n;
        }
    }
    if(x<0)
```



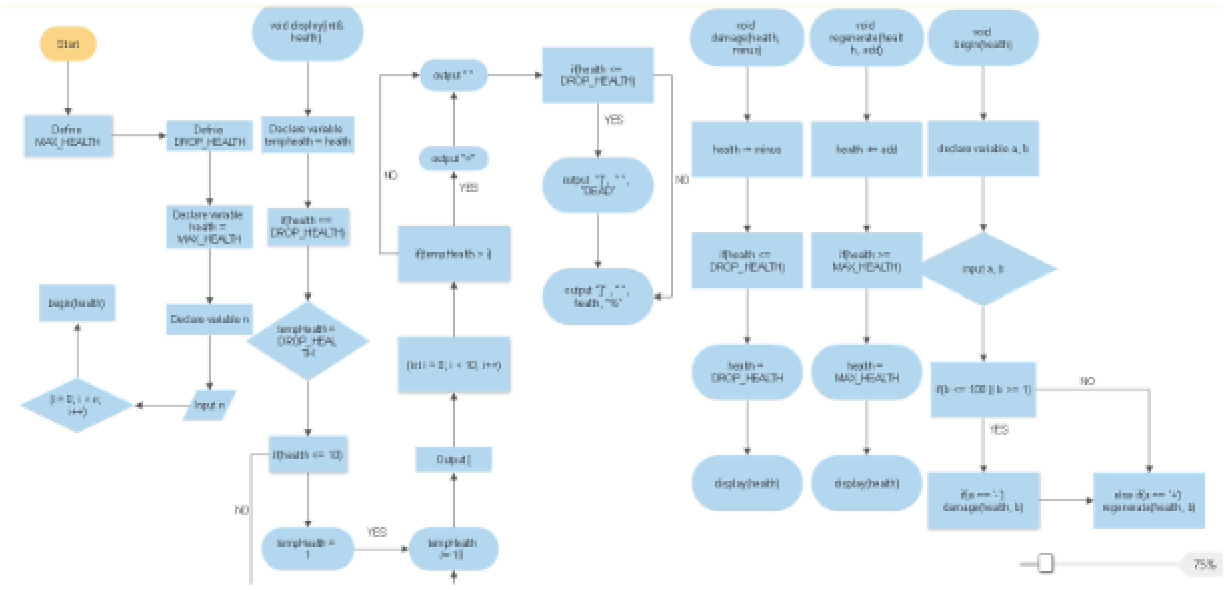
```
    {
        x=0;
    }
}


void regenerate (char &c, int &n, int &x)
{
    if(c=='+')
    {
        if(0<=n && n<=100)
        {
            x=x+n;
        }
    }
    if(x>100)
    {
        x=100;
    }
}


void display (char &c, int &n , int &x)
{
    if(x==0)
            {
                cout<<"[          ]"<<" "<<"DEAD"<<endl;
            }
            if(1<=x && x<10)
            {
                cout<<"[=         ] "<<x<<'%'<<endl;
            }
            else if(10<=x && x<20)
            {
                cout<<"[=         ] "<<x<<'%'<<endl;
            }
            else if(20<=x && x<30)
            {
                cout<<"[==        ] "<<x<<'%'<<endl;
            }
            else if(30<=x && x<40)
            {
                cout<<"[===       ] "<<x<<'%'<<endl;
            }
            else if(40<=x && x<50)
            {
                cout<<"[====      ] "<<x<<'%'<<endl;
            }
            else if(50<=x && x<60)
            {
                cout<<"[=====     ] "<<x<<'%'<<endl;
            }
```



```
            else if(60<=x && x<70)
            {
                cout<<"[======    ] "<<x<<'%'<<endl;
            }
            else if(70<=x && x<80)
            {
                cout<<"[=======   ] "<<x<<'%'<<endl;
            }
            else if(80<=x && x<90)
            {
                cout<<"[========  ] "<<x<<'%'<<endl;
            }
            else if(90<=x && x<=99)
            {
                cout<<"[========= ] "<<x<<'%'<<endl;
            }
            else if(x==100)
            {
                cout<<"[==========] "<<x<<'%'<<endl;
            }


}


int main()
{
    int N, n;
    char c;


    cin>>N;
    if(1<=N && N<=10)
    {
        int x=100;
        for(int i=0; i<N; i++)
        {
            cin>>c>>n;
            if((1<=n && n<=100) || (c=='+' && c=='-'))
            {
                if(c=='-')
                {
                    damage (c, n, x);
                }
                else if(c=='+')
                {
                    regenerate(c,n,x);
                }
            }
            display(c,n,x);
        }
    }
```

```
}
```

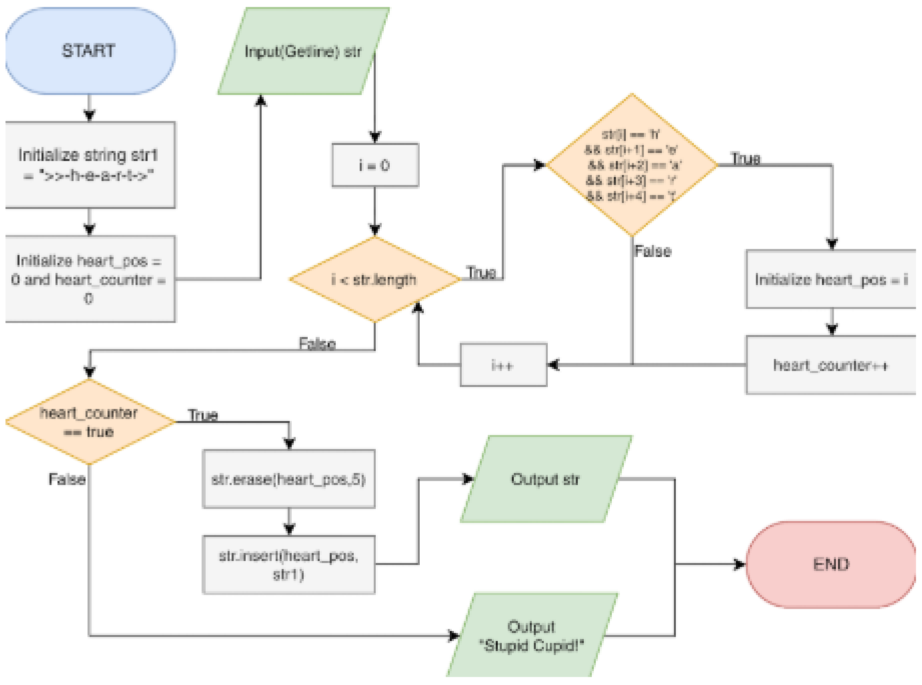
Flowchart



Stupid Cupid

```
#include <iostream>
#include <cstring>
using namespace std;

int main() {
    string S;
    while (getline(cin, S)) {
        bool found = false;
        string h = ">>-h-e-a-r-t->";
        string l = "heart";
        for (int i = 0; i < S.length(); i++) {
       if (S[i] == 'h'&& S[i +1 ] == 'e' && S[i +2] == 'a' && S[i+3] ==
'r' && S[i+4] == 't'){
               cout << S.substr(0, i) << h;
               found = true;
            S = S.substr(i + 5);
            }
        }
        if (found == true ) {
            cout << S << endl;
        } else {
            cout << "Stupid Cupid!" << endl;
        }
    }
    return 0;
}
```

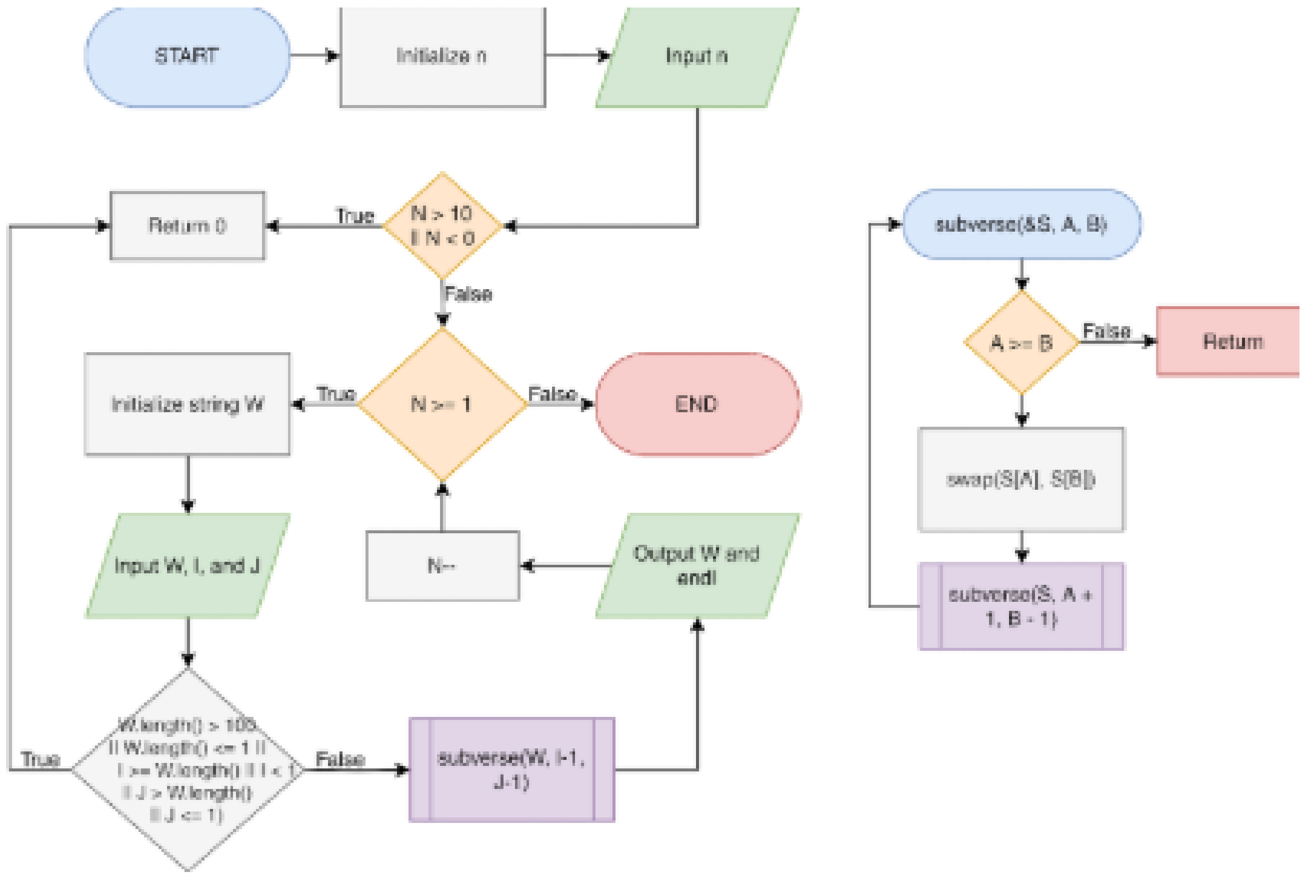
Flow chart


START
Input(Getline) str
Initialize string str1 = ">>-h-e-a-r-t->"
i = 0
str[i] == 'h' && str[i+1] == 'e' && str[i+2] == 'a' && str[i+3] == 'r' && str[i+4] == 't'
True
Initialize heart_pos = 0 and heart_counter = 0
i < str.length
True
False
Initialize heart_pos = i
False
i++
heart_counter++
heart_counter == true
True
False
str.erase(heart_pos,5)
Output str
str.insert(heart_pos, str1)
END
Output "Stupid Cupid!"

Substring Reverse

```
#include <iostream>
using namespace std;


string reverse(string w) {
  if (w.empty())
    return "";
  else
    return reverse(w.substr(1, w.length()-1)) + w.substr(0, 1);
}
int main() {
  int n;
  cin >> n;
  for (int x = 0; x < n; x++) {
    string w;
    int i, j;
    cin >> w >> i >> j;
    cout << w.substr(0,i-1 )<< reverse(w.substr(i-1, j-i+1))<<
w.substr(j) << endl;
  }
  return 0;
}
```

Flowchart



CASINO ROYALE\

```cpp
#include<iostream>
using namespace std;

int  royale (int N, int T, int e){
         if ( N == 0 ){
             cout << " ";
             return e;
         }
         if (T %2 == 0)
        {
          e = e + 3;
          T = T + 3;
          cout << "B";
        }
        else  {
          e = e - 1;
          T = T - 5;
          cout <<"A";
          }
    return royale ( N -1, T , e);

     }
int main(){
    int N,T,e;
    for ( int i = 0; i < 2 ; i++){
    cin >> N >> T;
    cout << royale (N,T,e) << "z" << endl;
    }
    return 0;
}
```

MAXIMUM PRODUCT

```cpp
#include<iostream>
using namespace std;

int main(){
    int T, n;
    int max = -100000, max2 = -100000;
    int M[n];
    cin >> T;
    if(T < 0 || T > 11)
        return 0;
    for(int j = 0;j < T;j++){
        cin >> n;
        if(n < 2 || n > 100000)
            return 0;
        for(int i = 0;i < n;i++){
            cin >> M[i];
        }
        int *ptr = &M[0];
        for(int i = 0;i < 5;i++){
            if(max < *ptr)
                max = *ptr;
            ptr++;
        }
        int *ptr2 = &M[0];
        for(int i = 0;i < 5;i++){
            if(max2 < *ptr2 && max > *ptr2)
                max2 = *ptr2;
            ptr2++;
        }
        cout << max*max2 << endl;
    }
    return 0;
}
```

<u>BUILD BUILD BUILD!</u>

```
#include<iostream>
#include <iomanip>
using namespace std;

int main(){
      int s, n;
      cin >> s;
      for (int i = 0; i < s; i++){
      double P;
      float height = 0.5;
      cin >> n >> P;
      int H[n];
      for (int j = 0; j < n; j++){
          cin >> H[j];
      }
      float max = 0;
      float earn = 0;
      for (int k = 0; k < n; k++){
          if (H[k] > height)
          {
              height = H[k];
               max ++ ;
               earn = max * P;
          }
      }
       cout <<fixed << setprecision(2) << earn <<  endl;
      }
          return 0;
      }
```

<u>IS IT RIGHT?</u>

```cpp
#include <iostream>
#include <cmath>

using namespace

struct point {
   int x, y;
};

struct Triangle {
   point p1, p2, p3;
};

bool pythagorean(point p1, point p2, point p3) {
   int a = pow((p1.x - p2.x), 2) + pow((p1.y - p2.y), 2);
   int b = pow((p2.x - p3.x), 2) + pow((p2.y - p3.y), 2);
   int c = pow((p3.x - p1.x), 2) + pow((p3.y - p1.y), 2);

   if (a == b + c || b == a + c || c == a + b) {
       return true;
   } else {
       return false;
   }
}

int main() {
   int c;
   cin >> c;
   for (int i = 0; i < c; i++) {
       Triangle t;
       cin >> t.p1.x >> t.p1.y >> t.p2.x >> t.p2.y >> t.p3.x >> t.p3.y;

       if (pythagorean(t.p1, t.p2, t.p3)) {
           cout << "YES" << endl;
       } else {
           cout << "NO" << endl;
       }
   }

   return 0;
}
```