Author: Elisa Ramos Gómez

Relational Operators

Note: the color in yellow means the result.

1. Started to cin my variable testcases.
2. Declare and cin two variables A and B
3. Compare A and B and cout the result if A '<' B , A '>' B, A '=' B

Code:

```
void comparer(long long a, long long b)
{
        if (a<b)cout<<"<\n";
         else if (a>b)cout<<">\n";
         else cout<<"=\n";

}
Int main(){
      Int test;
      Cin >> test;
      While(test--){
            Long long a,b;
            Cin >> a >> b;
            Comparer(a,b);
      }

}
```

Complexity: O(c) because I have to make the function comparer at most 15 times, so that's constant in time, and in memory because I only used 3 variables test, a and b.

Author: Elisa Ramos Gómez

Division of Nlogonia

1. Started to cin n cases and numbers of coordinates while n cases were different to 0 zero.
2. Cin n and m that was my mark of reference
3. Then cin x and y for all the range of 0 to n cases
   3.1. Compere x == n or y == m print "divisa"
   3.2. x > n and y > m print "NE"
   3.3. x > n and y < m print "SE"
   3.4. x < n and y > m print "NO"
   3.5. x < n and y < m print "SO"

Code:

```
int nCases;
int n,m,x,y,a;
int main()
{
ios_base::sync_with_stdio(0);
cin.tie(0);
while(cin>>nCases)
{
    cin>> n >> m;
    for(int i=0; i<nCases; i++)
    {
        cin>>x >>y;
        if(x==n||y==m){cout<<"divisa"<<endl;}
        else if(x>n && y>m){cout<<"NE"<<endl;}
        else if(x>n && y<m){cout<<"SE"<<endl;}
        else if(x<n && y>m){cout<<"NO"<<endl;}
        else{cout<<"SO"<<endl;}
    }
    if(nCases==0){break;}
}
 return 0;
}
```

Complexity: O (n^2) because n cases can be at most 10^3 so if I received the worst case and I used a for that means that I have to make de comparison in a range of 0 to 10^3 - 1 that means that I do all the operation 10^3 * 10^3 = 10^6 [multiply because exist the reset] that enter perfect in 1 second.

Author: Elisa Ramos Gómez

Cost Cutting

1. Started to cin my test cases
2. Declare an array with name data, then my variables a, b, c which are the numbers of comparison and my variable answer.
3. Iterate with a for in a range of 1 to test cases
4. Cin a, b, c
5. Compare (a > b and a < c) or (a < b and a > c) my answer = a
   5.1. Compare (b > a and b < c) or (b < a and b > c) my answer = b
   5.2. Otherwise answer = c
6. Save data[pos of my for] = answer
7. Make another for (x) to print "Case x:" << data[x] << endl;

Code:

```
int dato[20];
int n,a,b,c,res;
int main()
{
    cin>>n;
    if(n<=20)
    {
        for(int i=1; i<=n;i++)
        {
            cin>> a >> b >> c;
            if((a>b && a<c)||(a<b && a>c)){res=a;}
            else if((b>a && b<c)||(b<a && b>c)){res=b;}
            else{res=c;}
            dato[i]=res;
        }

        for(int i=1; i<=n;i++)
        {
            cout<< "Case "<< i<< ":" << " "<<dato[i]<< endl;
        }

    }
    return 0;
}
```

Complexity: O (2n) = O (n) because I made linear lecture, comparison and also save the answer in an array, and made another for to iterate my array and print my cases that at most my test would be 20 so this solution is linear but the value of N = 20 I can consider that is something constant for this specific range 1 < N <= 20. [20 < 10^6]

Author: Elisa Ramos Gómez

TEX Quotes

1. Getline all the text
2. Save in a data type string (string is the set of chars)
3. Iterate all the text, have a flag = true, made a string 1 with `` and string 2 with ''
4. The when I found "double quotes y change for string 1 with `` and made the flag = false
5. When I found "double quotes again and my flag = false now change to my string 2 and made flag = true again

Code:

```
string texto;
string str1("``");
string str2("''");
int main()
{
   bool primero=true;
    while(getline (cin,texto))
    {
            for(int i=0;i<texto.size();i++)
            {
                  if(texto[i]=='"')
                  {
                        if(primero)
                        {
                            texto.erase(i,1);
                            texto.insert(i,str1);
                            primero=false;
                        }
                        else
                        {
                            texto.erase(i,1);
                            texto.insert(i,str2);
                            primero=true;
                        }
                  }
            }
            cout<<texto<<endl;
   }
 return 0;
}
```

Complexity: O (n) because I have to iterate for al the size of the text and I depend of this. Another solution is cin each char and when I found double quotes print string 1 or 2 and use the flag to maintain the sequence without save it in another string.

Author: Elisa Ramos Gómez

Celebrity jeopardy

I. Started to cin each char and print at same time.

Code:

```
int ecuacion;
int main()
{
 do {
    ecuacion=getchar();
    putchar (ecuacion);
  } while (ecuacion != -1);
  return 0;

}
```

Complexity: O (C)
Getchar can read from standard input. It's equivalent to getc(stdin) and the return type is int to accommodate for the special value EOF that's why we can see inside code -1 that means EOF.
Putchar on success, the character written is returned, If a writing error occurs, EOF is returned and the error indicator (ferror) is set.

Author: Elisa Ramos Gómez

Combination Lock

1. Cin my variables number start, a, b, c.
2. Then if all these variables are different to 0.
3. Make the sum of the degrees of the movements of the lock
   - 3.1. Consider a constant of 1080 degrees because you must do 3 completely rounds of 360 degrees. 360 * 3 = 1080
   - 3.2. Then made the calculation of the first change that are de subtraction of my position a minus start, but we have to consider that we have inside the lock 40 elements, so you have to add this 40 and made module of 40 elements. (Start – a *+ 40) % 40* this are the technique to obtain only the degrees of the movement 30 – 0 = 30 + 40 = 70 mod 40 = 30 degrees
   - 3.3. Made the same subtraction of b – a use the especial technique
   - 3.4. Repeat the step 3.3 the subtraction of b – c and use the technique.
4. Made the sum of the result of 3.1 to 3.4 and multiply by 9 to obtain the total degrees because the degrees for each mark is 360/40 = 9 degrees for each one.

Code:

```
int inicio, a,b,c,res;
int main()
{
    while(cin >> inicio)
    {
        cin >> a >> b >> c;
        if(inicio==0 && a ==0 && b==0 && c==0){break;}
        res=(1080+((inicio-a+40)%40+ (b-a+40)%40 + (b-c+40)%40)*9);
        cout << res << endl;

    }

return 0;
}
```

Complexity: O (n) because depending the amount of how many times you have to make the formula but for the formula it's something constant, so that supports a lot of length of numbers and adequate for x - elements inside the lock.

Author: Elisa Ramos Gómez

Searching for Nessy

1. Cin my variable test cases
2. While test cases are different to 0 made the next steps
3. Cin my two variables n and m
4. Check if n and m are divisible in 3 use the formula (n*m) /9 that means the area total and how many sonars we can have and the area of them are 9
5. If n or m or n and m are not divisible exact by 3 you must recalculate n or m or both to calculate the cases bases of the numbers divisible by 3.

Code:

```
int main(){
    int test_cases;
    cin >> test_cases;
    while(test_cases--){
        long long n, m;
        cin >> n >> m;
        long long res = 0;
        long long area_rec = 0;
        area_rec = n*m;
        res = area_rec /9;

        cout <<area_rec <<endl;
        cout <<res <<endl;
        if(n % 3 != 0){
            int aux = 0;
            for(int i = 6 ; i < n ; i +=3){
                if(i == 6){aux = i;}
                else if(n > i){aux += 3;}
            }
            n = aux;
            area_rec = n*m;
            res = area_rec /9;
        }
         if(m % 3 != 0){
            int aux = 0;
            for(int i = 6 ; i < m ; i +=3){
                if(i == 6){aux = i;}
                else if(m > i){aux += 3;}
            }
            m = aux;
            area_rec = n*m;
            res = area_rec /9;
        }
        cout <<res <<endl;
    }
}
```

Complexity: O (n) because we have to made the different operations depending of the n cases, and the operations inside are constant. Another way to solve it's to made answer = (n/3) * (m/3).

Author: Elisa Ramos Gómez

Packing in holiday

1. Cin my variable test cases
2. do the next steps while test cases different of 0.
3. Cin the variables l, w, h that are the measures
4. If l, w and h are lower or equal to 20 made the counter = 3
5. If step for its true cout the case x = good
6. Otherwise cout the case x: bad

Code:

```
int t, l,w,h,suma,cont;
int main()
{
cont=1;
    cin >> t;
    while(t--)
    {
        suma=0;
        cin >>l >>w >>h;
        if(l <= 20){suma+=1;}
        if(w <= 20){suma+=1;}
        if(h <= 20){suma+=1;}
        if(suma==3)
        {cout << "Case "<<cont<<": "<<"good"<<endl;}
        else {cout << "Case "<<cont<<": "<<"bad"<<endl;}
        cont++;
    }
return 0;
}
```

Complexity: O (n) because the comparisons you have to make the n cases that the user give in this specific problem at most we have 100 cases so that you can considered that the real complexity is O (c) because 100 < 10^6.

Author: Elisa Ramos Gómez

Automatic Answer

1. Cin my variable test cases
2. Cin the variable n
3. Made the formula n=((((((n*567)/9) +7492) *235) /47) -498)
4. Decompound the variable n and print the place of the tens column

Code:

```
long long a,b,c,d,e,z;
int n;
void descomponer(long long z)
{
    a=z/100000;
    b=(z-(a*100000))/10000;
    c=(z-(a*100000+b*10000))/1000;
    d=(z-(a*100000+b*10000+c*1000))/100;
    e=(z-(a*100000+b*10000+c*1000+d*100))/10;
    cout<< e << endl;
}
int main()
{
ios_base::sync_with_stdio(0);
cin.tie(0);
cin>> n;
    while(n--)
    {
        cin >> z;
        z=((((((z*567)/9)+7492)*235)/47)-498);
        z=abs(z);
        descomponer(z);
    }
return 0;
}
```

Complexity: O (n) because you have to made the formula and the decompound the n times that the user requires, at most we received 100 test cases so the complexity of all is O (c), because we only made formulas 100 times.

Author: Elisa Ramos Gómez

Language Detection

1. Cin the string while the string was different of " # "
2. Make the comparison of the string and detect with language are and print the language

Code:

```
string cadena;
string fin("#");
string a("HELLO"); //ENGLISH
string b("HOLA"); // SPANISH
string c ("BONJOUR"); // FRENCH
string d ("HALLO"); //GERMAN
string e ("CIAO"); //ITALIAN
string f ("ZDRAVSTVUJTE"); //RUSSIAN
int i=1;
int main()
{
ios_base::sync_with_stdio(0);
cin.tie(0);

    while(cin >> cadena)
    {
        if(cadena== fin){break;}
        else if(cadena.compare(a)== 0){cout<< "Case "<< i<< ":" << " "<<"ENGLISH"<<
endl; i++;}
        else if(cadena.compare(b)== 0){cout<< "Case "<< i<< ":" << " "<<"SPANISH"<<
endl; i++;}
        else if(cadena.compare(c)== 0){cout<< "Case "<< i<< ":" << " "<<"FRENCH"<<
endl; i++;}
        else if(cadena.compare(d)== 0){cout<< "Case "<< i<< ":" << " "<<"GERMAN"<<
endl; i++;}
        else if(cadena.compare(e)== 0){cout<< "Case "<< i<< ":" << " "<<"ITALIAN"<<
endl; i++;}
        else if(cadena.compare(f)== 0){cout<< "Case "<< i<< ":" << " "<<"RUSSIAN"<<
endl; i++;}
        else{cout<< "Case "<< i<< ":" << " "<<"UNKNOWN"<< endl; i++;}
    }
return 0;
}
```

Complexity: O (n) depending of the number of cases that requires, we have to do the comparison and print at most 2 * 10^3, the function compare()the complexity are unspecified, but generally up to linear in both the compared and comparing strings lengths. Other way to solve it's to compare the first and second char of all the string.

Author: Elisa Ramos Gómez

Watermelon

1. Cin the int variable watermelon
2. Check if watermelon is multiple of 2 print Yes, except if watermelon == 2
3. Otherwise print No

Code:

```
Int main(){
      Int watermelon;
      Cin >> watermelon;
      If(watermelon == 2){
            Cout <<”No\n”;
      }else if(watermelon % 2 == 0){
            Cout <<”Yes\n”;
      }else{
            Cout <<”No\n”;
      }
Return 0;
}
```

Complexity: O (c) because the number at most will be 100 and you only made 3 comparisons and print the answer.

Author: Elisa Ramos Gómez

Way Too Long Words

1. Cin the test cases
2. While test cases different 0 made the next steps
3. Cin the string
4. Compare if the size of the string are lower or equal than 10 print the string
5. If not made a counter and count in the range of the second letter to size-1 print first char, the counter and the last char of the string.

Code:

```
void solve(){
    string s;
    cin >> s;

   if(s.size() <= 10){
    cout << s <<"\n";
   }else{
        int ct = 0;
        for(int i = 1; i < s.size()-1 ; i++){
            ct++;
        }
        cout <<s[0] <<ct <<s[s.size()-1]<<"\n";
   }
}
int main(){
    int t;
    cin >> t;
    while(t--){
        solve();
    }
return 0;
}
```

Complexity: O (n*s.size()) = O (100*100) = O(10^4) because the function solve made the iterator for all the size and in the worst case the size will be 100 and if we have n = 100 the multiply because the for made like an reset for each case.

Author: Elisa Ramos Gómez

Team

1. Cin the test cases
2. Declare the variables answer, a, b, c.
3. For each test cin a, b, c
4. Made the comparison if the sum of a+b+c >= 2 add in one the counter answer
5. Print answer at final of the while.

Code:

```
int main(){
    int numeros;
    cin >> numeros;
    int respuesta = 0;
    while(numeros--){
        int a, b, c;
        cin >> a >> b >> c;

        if((a+b+c) >= 2){
            respuesta++;
        }

    }
    cout << respuesta << endl;
return 0;
}
```

Complexity: O (n) at most n = 10^3 because we only use 5 variables and make constant operations for each case.

Author: Elisa Ramos Gómez

Domino piling

1. Cin the variables m and n
2. Do the variable answer equal to the formula ((m*n) - 1) / 2
3. And at final if the division you take x.5 made function ceil answer = 2.5 + 0.5 = 3
4. Print answer

Code:

```
int main(){
    double m ,n;
    cin >> m >> n;
    double res = 0;
    res = ((m*n) - 1)/2;
     if(res - int(res) == 0.5)res +=0.5;
    cout << res <<  endl;
return 0;
}
```

Complexity = O (C) because n and m at most the value will be 16 and you only made a formula and operations that are constant.

Author: Elisa Ramos Gómez

Next Round

1. Cin the variables n and k
2. Cin a1, a2, ..., an. To save an array, declare a counter
3. Then made one variable auxiliar to use like and pivot in the position k – 1
4. Iterate for all the array and count the numbers that are greater or equal than the auxiliary variable and check if both auxiliary and the ai of the array are different to 0.
5. Print counter

Code:

```
int main(){
    int  n,k;
    cin >> n >> k;
    int aux = 0;
    int res = 0;
    int score[n+2];
    for(int i = 0; i < n ; i++) cin >> score[i];
    aux = score[k-1];
    for (int i = 0; i < n ; i++){
        if(score[i] >= aux && (score[i]!=0 || aux != 0)){
            res++;
        }
    }
    cout <<res<<endl;
return 0;
}
```

Complexity = O (n) because n at most the value was 50 so we consider that will be constant because we only made constant comparisons.