```
////////////////////////////
Recovering a string  c++
////////////////////////////

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

int main()
{
    int t;
    cin >> t;
    while (t--){
        int n;
        cin>>n;
        string ans="";
        int i =1;
        while (n-i>52)
            i++;
        ans +=('a'+i-1);
        n-=i;
        i=1;
        while (n-i>26)
        i++;
        ans +=('a'+i-1);
        ans +=('a'+n-i-1);
        cout<<ans<<endl;
    }
    return 0;
}
```

```
/////////////////////////////
Recovering a string  c
/////////////////////////////
#include <stdio.h>

int main()
{
    int t;
    scanf("%d", &t);
    while (t--)
    {
        int n;
        scanf("%d", &n);
        char ans[4]; // Assuming the answer will have at most 3 characters

        int i = 1;
        while (n - i > 52)
            i++;
        ans[0] = 'a' + i - 1;
        n -= i;

        i = 1;
        while (n - i > 26)
            i++;
        ans[1] = 'a' + i - 1;
        ans[2] = 'a' + n - i - 1;
        ans[3] = '\0'; // Null-terminate the string

        printf("%s\n", ans);
    }
    return 0;
}
```

```
//////////////////////////
Road and gang c
/////////////////////////

#include <stdio.h>

void solve() {
    int t; // Number of test cases
    scanf("%d", &t);
    // Loop through each test case
    while (t--) {
        int n; // Number of districts
        scanf("%d", &n);

        int a[n]; // Array to store the gang each district belongs to
        int unique_district = -1; // This will store the first district with a unique gang

        // Read in the district gangs
        for (int i = 0; i < n; i++) {
            scanf("%d", &a[i]);
        }
        // Check if all districts are in the same gang
        int first_gang = a[0]; // Initial gang to compare with others
        int all_same_gang = 1; // Flag to check if all districts are in the same gang

        for (int i = 1; i < n; i++) {
            if (a[i] != first_gang) {
                all_same_gang = 0; // Found a different gang
                unique_district = i + 1; // Index of the unique gang (1-based)
                break;
            }
        }

        if (all_same_gang) {
            // If all districts are in the same gang, it's impossible to create n-1 roads
            printf("NO\n");
        } else {
            // If there is at least one unique district, we can build the roads
            printf("YES\n");

            // Connect the first district with all other districts
            for (int i = 2; i <= n; i++) {
                printf("1 %d\n", i); // Create a star topology
            }
        }
    }
}

int main() {
    solve(); // Call the function to solve the problem
    return 0;
}
```

```
//////////////////////////////////
```

3. B. Rudolf and 121 (Row 0)in c

```
//////////////////////////////////

#include <iostream>
#include <algorithm>

using namespace std;

bool canMakeZero(long long* arr, int n) {
    for (int i = n - 2; i >= 1; --i) {
        if (arr[i] != 0) {
            long long operations = arr[i];

            if (arr[i - 1] < operations || arr[i + 1] < operations) {
                operations = min(arr[i - 1], arr[i + 1]);
            }

            arr[i - 1] -= operations;
            arr[i] -= 2 * operations;
            arr[i + 1] -= operations;
        }
    }

    // After processing, check if the array is zeroed
    return arr[0] == 0 && arr[n - 1] == 0;
}

int main() {
    int t;
    cin >> t; // Read the number of test cases

    while (t--) {
        int n;
        cin >> n; // Read the size of the array

        long long* arr = new long long[n];
        for (int i = 0; i < n; ++i) {
            cin >> arr[i]; // Read the elements of the array
        }

        // Attempt to zero out the array using the defined operation
        if (canMakeZero(arr, n)) {
            cout << "YES" << endl;
        } else {
            cout << "NO" << endl;
        }

        delete[] arr; // Free the dynamically allocated memory
    }

    return 0;
}
```

///////////////////////////////
Increase and Copy c++
////////////////////////////////

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

int main()
{
    long long t,n,k,i;
    cin>>t;
    while(t--)
    {
        cin>>n;
        long long Min=9999999999999;
        if(n==1)
        {
            cout<<"0"<<endl;
            continue;
        }
        for (i=2;i<500000;i++)
        {
            if(n%i==0)
                k=(n/i)-1+(i-1);
            else
                k=(n/i)+(i-1);
            Min=min(Min,k);
        }
        cout<<Min<<endl;
    }
    return 0;
}


//////////////////////////////////
Clock
//////////////////////////////////

#include <stdio.h>
#include <stdlib.h>

int cmp(int a, int b)
{
    int num[4];
    num[0]=a/10;
    num[1]=a%10;
    num[2]=b/10;
    num[3]=b%10;
    for(int i=0;i<4;i++)
    {
        if(num[i] !=num[3-i])
        {
            return 0;
        }
    }
    return 1;
}

int main(void)
{
   int n,hour,min,num,a,b;
   scanf("%d",&n);
   for(int i=0;i<n;i++)
   {
       int count=0, flag=0;
       scanf("%d:%d %d",&hour,&min,&num);
       a=hour;
       b=min;
       do
       {
           a+=num/60;
           b+=num%60;
           if(b>=60)
```

```
            {
                b=b%60;
                a++;
            }
            a=a%24;
            if(cmp(a,b))
            {
                count++;
            }
        }
        while(a!=hour || b!=min);
        printf("%d\n",count);
    }
    return 0;
}
```

//
Stack… AAAABBBBABA
//

```
#include <iostream>
#include <bits/stdc++.h>
#include <stack>

using namespace std;

int main()
{
    int t;
    cin>>t;
    while(t--)
    {
        string s;
        cin>>s;
        stack<char>st;

        for(int i=0;i<s.size();++i)
        {
            if(!st.empty())
            {
                if((s[i]=='B' && st.top()=='A') || (s[i]=='B' && st.top()=='B'))
```

```
                    st.pop();
                else
                    st.push(s[i]);
            }
            else
                st.push(s[i]);
        }
        cout<<st.size()<<"\n";
    }
    return 0;
}
```