**ADVANCED ALGORITHMS AND APPLICATIONS**

**PTIT 2023**

|  |  |
| --- | --- |
| A person in a suit standing outside  Description automatically generated with low confidence | Lecturer  PGS TSKH Nguyễn Xuân Huy  Viện Công nghệ Thông tin  Viện Hàn lâm Khoa học  và Công nghệ Việt nam  [nxhuy564@gmail.com](mailto:nxhuy564@gmail.com)  0903203800 |

Zalo Samsung23 Trao doi

Contents

[**Information** 3](#_Toc138961879)

[**Schedule** 3](#_Toc138961880)

[Introduction 3](#_Toc138961881)

[**Input for testing** 3](#_Toc138961882)

[Triangle Numbers 3](#_Toc138961883)

[Pentagonal numbers 5](#_Toc138961884)

[Hexagonal numbers 5](#_Toc138961885)

[Magic Squares 5](#_Toc138961886)

[E5 Five examples 5](#_Toc138961887)

[Odd Magic Square (OMS) 6](#_Toc138961888)

[Using stringstrean for Str 8](#_Toc138961889)

[Even Magic Squere EMS 8](#_Toc138961890)

[**Joshephus Problem (game)** 8](#_Toc138961891)

[If you want...Ver 6 11](#_Toc138961892)

[One more...Ver 7: Simplest queue 12](#_Toc138961893)

[MyQueue Again 14](#_Toc138961894)

[Using stringstream 15](#_Toc138961895)

[**Rectangle Cipher** 16](#_Toc138961896)

[A new Version 17](#_Toc138961897)

[Farey Sequences 19](#_Toc138961898)

[Version 1 19](#_Toc138961899)

[Version 2 21](#_Toc138961900)

[Version 3 23](#_Toc138961901)

[Roman Numerals 25](#_Toc138961902)

[Expression 28](#_Toc138961903)

[Simple version 28](#_Toc138961904)

[Expression with Numbers 31](#_Toc138961905)

[The next steps 34](#_Toc138961906)

**Information**

## **Schedule**

|  |  |  |  |
| --- | --- | --- | --- |
| **NN** | **Contents** | **Theory** | **App** |
| 1 | Complecity of Algorithms |  |  |
| 2 | Brute Force, Back tracking, Recursive, Divide and Conquer |  |  |
|  | *Contest 1 + Contest 2* |  |  |
| 3 | Greddy method |  |  |
| 4 | Dinamic programming |  |  |
|  | *Contest 3 + Contest 4* |  |  |
| 5 | Stack |  |  |
| 6 | Queue |  |  |
|  | *Contest 5 + Contest 6* |  |  |
| 7 | Sorting and Searsing |  |  |
| 8 | Graph Searching |  |  |
|  | *Contest 7 + Contest 8* |  |  |
| 9 | Graph with weight |  |  |
| 10 | Binary Trees |  |  |
|  | *Contest 9 + Contest 10* |  |  |
|  | **Total** |  |  |

**Doc**

<https://drive.google.com/drive/folders/1KZ5OSm693i2aJe65JqDv9cW92M0-EO_b?usp=sharing>

<https://drive.google.com/drive/folders/1GCRp4lzwp_4tA8zBoXEw97JB-VigsBE1?usp=sharing>

## Introduction

Lecturer: 0903203800 nxhuy564@gmail.com. Hay quên, bấm máy kém

Students: Chăm, testing kém

Agreements:

\* Time*19:30 - 21:30*

\* Little but GOOD Tha it ma tot

\* Formats mau, Rules, Slogans, Recomments

Sample, Pattern, Examples

Why we have to learn programming?

Programs = Algorithms + Data Structures

Mam com = Cach nau + Nguyen lieu

**Problem** ?

* Understanding
* Algorithm
* Data Structures

|  |  |
| --- | --- |
| Algorithm | Data Structures |
| ABCDF | ABCDF |
| AA, AB, AC, AD, AF  BA, BC, BD, BF  CA, CB, … | |
| A: thuong thua, B: xuat sac  C: kha, D: co len, F: ? | |

Data structures: do dung = (Dung gi, dung luong, xu li)

int, long, long long, unsigned (int, long, long long)

char, short, float, double

string

file

array

vector

stringstream

file

stack

queue

**Cac thao tac**

Khoi tao

Gan x = y cung kieu. khac kieu, To = nho, nho = to

float = int ? int = float ?

xen, xoa, sua (update)

so sanh

**Nguyen ly: bat bien, thay doi**

int = (D, P)

D =

cout << "\n " << INT\_MIN << " " << (1<<31) << " " << (int)pow(2,31);

cout << "\n " << INT\_MAX << " " << ((1<<31) - 1) << " " << (int)pow(2,31)-1;

P:

+ - \* / %

+= -= \*= /= %=

<<k >>k <<=k >>=k

== != > < >= <=

& | ^ ~

&= |= ^= ~=

???

main() {

int a, b, z;

a = 2; b = 3;

cout << "\n 1. Init a = " << a << " b = " << b << " z = " << z;

z = Plus(a,b);

cout << "\n 1. z = Plus(a,b): a = " << a << " b = " << b << " z = " << z;

//--------------------

a = 2; b = 3;

cout << "\n 2. Init a = " << a << " b = " << b << " z = " << z;

a = Plus(a,b);

cout << "\n 2. a = Plus(a,b): a = " << a << " b = " << b << " z = " << z;

//--------------------------

a = 2; b = 3;

cout << "\n 3. Init a = " << a << " b = " << b << " z = " << z;

z = Plus(++a,b);

cout << "\n 3. z = Plus(++a,b): a = " << a << " b = " << b << " z = " << z;

//--------------------

a = 2; b = 3;

cout << "\n 4. Init a = " << a << " b = " << b << " z = " << z;

z = Plus(a++,b);

cout << "\n 4. z = Plus(a++,b): a = " << a << " b = " << b << " z = " << z;

//--------------------

a = 2; b = 3;

int \*v = &a;

cout << "\n 5. Init a = " << a << " b = " << b << " v = " << \*v;

\*v = Plus(a,b);

cout << "\n 5. \*v = Plus(a,b) a = " << a << " b = " << b << " v = " << \*v;

cout << endl << "\n\n T h e E n d \n";

return 0;

}

**Invariant**

**Thu cong** 2 dau thu A, B ✂

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | | |  | A | | |  | B | |  | A | |  | B |  | A |

A: chu nhat 🡪 vuong

B: vuong 🡪 CN

**Chocolat**

**Boc soi 1**

1 dong soi n vien, boc toi da k vien

n = 10; k = 3

**Boc soi 2**

boc 1 dong thoai mai

n = 10, m = 17

**Boc soi k**

Khi nao thi A di truoc?

**Nguyen ly thiet ke Algorithm (phuong phap)**

* Vet can (Brute force) \*
* Nhanh can \*\*
* Greedy \*\*
* Divide and conquer \*\*\*
* Dynamic programming \*\*\*
* Back tracking \*\*
* Dai so \*\*\*
* Genetic
* Ant colony
* Particle Swarm Optimization
* Deep leaning
* . . .

**Cac nguyen ly khac**

* Clean Room (IBM)
* A little but good
* 5E
* View and Rules
* Dream: I have a dream (ban than la nguoi su dung hhe thong)
* Tu phan bien
* Improve, Improve, and Improve (
* Tẩy não (thế mà mình cử tưởng!)
* Selfill

**Program Stucture in C++**

/\*

Name: Lesson1.CPP

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description:

Introduction

\*/

#include <bits/stdc++.h>

/\*

Ignore all

#include <iostream>

#include <fstream>

#include <windows.h>

#include <ctype.h>

#include <algorithm>

\*/

using namespace std;

string song[] = {"Happy Birthday...", "ABC", "We are nice friends", "!"};

#define SoChai int

typedef int DuDu, Ca;

typedef float Lac;

typedef float Rau;

typedef int BoKho;

typedef int Ca;

typedef int Bun;

void Go() {

cout << " ? ";

fflush(stdin);

if(cin.get() == '.') exit(0);

}

void Bia(SoChai c) {

string s = "Uong vua du nhe!";

cout << "\n " << s;

}

int Nom(DuDu d, Lac l, BoKho b) {

int result = d + l + b\*5;

cout << "\n Mon nom Hoang De: " << result;

return result;

}

int LauThai(Ca c, Rau r, Bun b) {

int result = c\*2 + r+r + b\*2;

cout << "\n Lau Thai: " << result;

return result;

}

void Music(int n) {

for(int i = 1; i <= 5; ++i) {

for(int j = 0; song[j] != "!"; ++j)

cout << "\n \*\*\* " << (char)14 << " " << song[j];

}

}

void SinhNhat() {

Bia(10);

Nom(20, 7, 12);

LauThai(3, 25, 7);

Music(3);

}

main() {

SinhNhat();

cout << endl << "\n\n T h e E n d \n";

return 0;

}

## **Input for testing**

### Triangle Numbers

*Triangle numbers are generated by the following formula:*

*T(n) = 1 + 2 + ... + n*

*For each integer x, write "Yes" if x is a triangle number; else write "No".*

*5E 5 vi du*

*15 7 21 0 1*

/\*

Name: Name: TP.CPP (Triangke number)

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description:

T(n) = 1 + 2 + ... + n = n(n+1)/2

\*/

#include <bits/stdc++.h>

using namespace std;

int TNum(int x) {

// T(n) = n(n+1) / 2 = x ?

// n(n+1) = 2x

int x2 = x + x;

int n = (int)sqrt(x2);

return (n\*(n+1) == x2) ? n : -1;

}

/\*-------------------------------

T(0) = 0

T(1) = 1

T(2) = 1+2 = 3

T(3) = 1 + 2 + 3 = 6

T(4) = T(3) + 4 = 10

T(5) = T(4) + 5 = 15

---------------------------------\*/

void Test1() { // input from an array

int x[] = {15, 99\*50, 7, 6\*11, 102, 0, 45, -1};

for(int i = 0; x[i] >= 0; ++i) {

int n = TNum(x[i]);

cout << "\n " << x[i] << " ? " << n;

if (n >= 0) cout << " Yes";

else cout << " No";

}

}

void Test2(string s) { // input from a string

stringstream ss(s);

int x;

while(true) {

ss >> x;

if (x < 0) break;

int n = TNum(x);

cout << "\n " << x << " ? " << n;

if (n >= 0) cout << " Yes";

else cout << " No";

}

}

void Test3(const char \* fn) { // input from a file

ifstream f(fn);

if(f.fail()) {

cerr << "\n File not found " << fn;

return;

}

int x;

while(true) {

f >> x;

if(x < 0) break;

int n = TNum(x);

cout << "\n " << x << " ? " << n;

if (n >= 0) cout << " Yes";

else cout << " No";

}

}

void Test4() { // input from keyboard

int x;

while(true) {

cout << "\n x = ";

fflush(stdin);

cin >> x;

if(x < 0) break;

int n = TNum(x);

cout << "\n " << x << " ? " << n;

if (n >= 0) cout << " Yes";

else cout << " No";

}

}

void Test5(int m, int range) { // random input

int x, r, n;

srand(time(NULL));

for(int i = 0; i < m; ++i) {

r = rand() % range;

x = r\*(r+1)/2;

n = TNum(x);

cout << "\n " << x << " ? " << n;

if (n >= 0) cout << " Yes";

else cout << " No";

x += 5;

n = TNum(x);

cout << "\n " << x << " ? " << n;

if (n >= 0) cout << " Yes";

else cout << " No";

}

}

void Test6(int d, int c) { // range testing

int n;

for(int x = d; x <= c; ++x) {

n = TNum(x);

cout << "\n " << x << " ? " << n;

if (n >= 0) cout << " Yes";

else cout << " No";

}

}

main() {

// Test1();

// Test2("15 4950 7 66 102 0 45 -1 ");

// Test3("TP.INP");

// Test4();

// Test5(10, 100);

Test6(40, 70);

cout << endl << "\n\n T h e E n d \n";

return 0;

}

Nhan xet

\* hieu bai ?

\* De / kho ? Nhanh / cham ?

\* Dieu nao la bo ich

### Pentagonal numbers

*Pentagonal numbers are generated by the formula n(3n-1)/2.*

*For each integer x, write "Yes" if x is a pentagonal number; else write "No".*

***Algorithm***

*P(n) = so ngu giac thu n*

*P(n) = x = n(3n-1)/2*

*Given x*

*Find n ?*

*Phuong trinh bac 2, nghiem n: , a = 3, b = -1, c = 2x*

***Complexity****: O(1)*

***Program***

/\*

Name: Name: PN.CPP (Pentagon number)

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description:

P(n) = n(3n-1)/2

\*/

#include <bits/stdc++.h>

using namespace std;

int PNum(int x) {

if (x < 0) return false;

if (x < 2) return true;

// P(n) = n(3n-1) / 2 = x ?

// n(3n-1) = 2x

// 3n^2 - n - 2x = 0, a = 3, b = -1, c = -2x, n ?

int d = 1+24\*x; // bb - 4ac

double n = (1 + sqrt(d)) / 6; // n chia het 8 ?

return (int(n) == n) ? n : -1;

}

/\*-------------------------------

n(3n-1)/2

P(0) = 0

P(1) = 1

P(2) = 5

P(3) = 12

P(4) = 22

P(5) = 35

...

---------------------------------\*/

void Test1() { // input from an array

int m = 10;

int x[m];

// Gen

for(int n = 0; n < m; ++n) {

x[n] = n\*(3\*n-1)/2;

}

for(int i = 0; i < m; ++i) {

int n = PNum(x[i]);

cout << "\n " << x[i] << " ? " << n;

if (n >= 0) cout << " Yes";

else cout << " No";

int y = x[i] + 3;

n = PNum(y);

cout << "\n " << y << " ? " << n;

if (n >= 0) cout << " Yes";

}

}

// Test2 .. Test6

main() {

Test1();

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### Hexagonal numbers

*Hexagonal numbers are generated by the formula n(2n-1).*

*For each integer x, write "Yes" if x is a hexagonal number; else write "No".*

## Magic Squares

Square nxn number 1..n2 all sums (row, colums, diagonals) are equals = Charcteristic Number

(dac so)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S3 | | |  | S5 | | | | |  | S4 | | | |  | S6 | | | | | |  | C = (1+..+n2)/n  = (n2+1)\*n2 / (2n)  = (n2+1)\*n / 2  C2 = 15, C4 = 35  C5 = 65, C6 = 111 |
| 8 | 1 | 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | 5 | 7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | 9 | 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

### E5 Five examples

|  |  |
| --- | --- |
| Algorithm OMS (Odd Magic Squere) |  |
| 1. set all 0 to a  2. write 1 at (0, n/2)  3. for k = 2..n2:  3.1 Find cell a(i,j)  3.2 set a(i,j) = k  Go to direction North-East (NE) |

### Odd Magic Square (OMS)

Program

/\*

Name: MS.CPP

Copyright: (C) 2023

Author: Devcpp Fan

Date: 22-06-23 11:50

Description: Magic Square

'''

\*/

#include <bits/stdc++.h>

using namespace std;

const int MN = 100;

int a[MN][MN];

int maxlen;

const string DIGIT = "0123456789";

void Go() {

cout << " ? ";

fflush(stdin);

if (cin.get()=='.') exit(0);

}

string Str(int n) {

if(n == 0) return "0";

string s = "";

while(n) {

s = DIGIT[n % 10] + s;

n /= 10;

}

return s;

}

void Print(int a[], int n, const char \* msg = "") {

cout << msg;

for (int i = 0; i < n; ++i) {

string s = Str(a[i]);

for(int j = s.length(); j <= maxlen; ++j) {

s = " " + s;

}

cout << " " << s;

}

}

void Print(int a[][MN], int n, const char \* msg = "") {

cout << msg;

for (int i = 0; i < n; ++i) {

Print(a[i], n, "\n ");

}

}

void Test(int n) {

int c = (n\*n+1)\*n/2;

int col, row;

int d1 = 0, d2 = 0;

int er = 0;

for(int i = 0; i < n; ++i) {

col = row = 0;

d1 += a[i][i]; d2 += a[i][n-1-i];

for(int j = 0; j < n; ++j) {

col += a[i][j];

row += a[j][i];

}

if(col != c) {

cout << "\n ERROR in row " << i;

++er;

}

if(row != c) {

cout << "\n ERROR in col " << i;

++er;

}

}

if(d1 != c) {

cout << "\n ERROR in diagonal 1";

++er;

}

if(d2 != c) {

cout << "\n ERROR in diagonal 2";

++er;

}

if(er == 0) cout << "\n CORRECT.";

}

void OMS(int n) {

for(int i = 0; i < n; ++i)

for(int j = 0; j < n; ++j)

a[i][j] = 0;

int i = 0, j = n/2;

a[i][j] = 1;

int nn = n\*n;

for(int k = 2; k <= nn; ++k) {

// (i,j) ?

--i; ++j;

if(i < 0 && j == n) {

// below (i,j)

i += 2; --j;

}

else if(i < 0) i = n-1;

else if(j == n) j = 0;

if(a[i][j] > 0) {

i += 2; --j;

}

a[i][j] = k;

}// for

Print(a,n,"\n Result: \n");

Test(n);

}

void EMS(int n) {

cout << "\n Will be released.";

}

void MS(int n) {

cout << "\n Magic Square of " << n;

if(n < 1 || n == 2) {

cout << "\n n < 0 or n = 2: No solutions.";

return;

}

if(n == 1) {

cout << "\n Result: 1";

return;

}

maxlen = Str(n\*n).length();

cout << "\n maxlen = " << maxlen;

if (n % 2 == 1) OMS(n);

else EMS(n);

}

main() {

for(int n = -3; n < 20; ++n) {

MS(n); Go();

}

// cout << Str(102030) << " " << Str(0);

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### Using stringstrean for Str

string Str(int n) {

stringstream ss;

ss << n;

string s;

ss >> s;

return s;

}

### Even Magic Squere EMS

## **Joshephus Problem (game)**

Rounding of n elements, count k elem. Who is the rest ?

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   m = 14, k = 3, res = 2 |

Algorithm? Data Structure?

5 approaches

Version 1. rounding scan (rai gianh) int array

Version 2. rounding scan (rai gianh) bit array

Version 2. rounding scan (rai gianh) bit array

/\*

Name: JP.CPP

Copyright: (C) 2023

Author: Devcpp Fan

Date: 22-06-23 11:50

Description: Josephus Problem

'''

\*/

#include <bits/stdc++.h>

using namespace std;

void Go() {

cout << " ? ";

fflush(stdin);

if (cin.get()=='.') exit(0);

}

void Print(int a[], int d, int c, const char \* msg = "") {

cout << msg;

for (int i = d; i <= c; ++i) {

cout << " " << a[i];

}

}

void Print(char b[], int d, int c, const char \* msg = "") {

cout << msg;

for (int i = d; i <= c; ++i) {

cout << " " << (int)b[i];

}

}

// int array, mark 0/1 O(n)

int JP1(int n, int k) {

int a[n];

for(int i = 0; i < n; ++i) a[i] = i+1;

int n1 = n-1;

Print(a, 0, n1, "\n Init");

int d, j = -1;

for(int i = 1; i < n; ++i) {

d = 0;

while(d < k) {

j = (j + 1) % n;

if(a[j] > 0) ++d;

}

a[j] = 0;

Print(a, 0, n1, "\n a:");

} // i

Print(a, 0, n-1, "\n Rest:");

int res;

for(int i = 0; i < n; ++i)

if (a[i] > 0) {

res = a[i];

break;

}

cout << "\n Result: " << res;

return res;

}

// int array, counting O(n)

int JP2(int n, int k) {

int a[n];

for(int i = 0; i < n; ++i) a[i] = 1;

int n1 = n-1;

Print(a, 0, n1, "\n Init");

int d, j = -1;

for(int i = 1; i < n; ++i) {

d = 0;

while(d < k) {

j = (j + 1) % n;

d += a[j];

}

a[j] = 0;

Print(a, 0, n1, "\n a:");

} // i

Print(a, 0, n-1, "\n Rest:");

int res;

for(int i = 0; i < n; ++i)

if (a[i] > 0) {

res = i+1;

break;

}

cout << "\n Result: " << res;

return res;

}

// bit array, O(n)

int JP3(int n, int k) {

char a[n];

for(int i = 0; i < n; ++i) a[i] = 1;

int n1 = n-1;

Print(a, 0, n1, "\n Init");

int d, j = -1;

for(int i = 1; i < n; ++i) {

d = 0;

while(d < k) {

j = (j + 1) % n;

d += a[j];

}

a[j] = 0;

Print(a, 0, n1, "\n a:");

} // i

Print(a, 0, n-1, "\n Rest:");

int res;

for(int i = 0; i < n; ++i)

if (a[i] > 0) {

res = i+1;

break;

}

cout << "\n Result: " << res;

return res;

}

void Print(vector<int> a, const char \* msg = "") {

cout << msg;

for (int i = 0; i < a.size(); ++i) {

cout << " " << a[i];

}

}

// Queue O(n)

int JP4(int n, int k) {

vector<int> q;

for(int i = 1; i <= n; ++i)

q.push\_back(i);

Print(q, "\n Init: ");

for(int i = 1; i < n; ++i) { // lap n-1 lan

// chuyen k-1 phan tu dau ve cuoi

for(int j = 1; j < k; ++j) {

q.push\_back(q.front()); // lay dau

q.erase(q.begin()); // bo dau

} // j

q.erase(q.begin()); // bo phan tu thu k

Print(q, "\n q: "); //Go();

} // i

int res = q.front();

cout << "\n Result: " << res;

return res;

}

void Rev(int a[], int d, int c) {

while(d < c) {

int x = a[d]; a[d] = a[c]; a[c] = x;

++d; --c;

}

}

// 123|4567 -> 4567|123

// 321|7654 -> 4567|123

// chuyen m phan tu ve cuoi

// 3n

void Move(int a[], int n, int m) {

Rev(a, 0, m-1);

Rev(a, m, n-1);

Rev(a, 0, n-1);

}

// Myqeueu, O(n) Tu cai queue

int JP5(int n, int k) {

int a[n];

for(int i = 0; i < n; ++i) a[i] = i+1;

Print(a, 0, n-1, "\n Init: ");

int nn = n;

for(int i = 1; i < nn; ++i) { // lap n-1 lan

Move(a, n, k);

--n;

Print(a, 0, n-1, "\n a: ");

}

return a[0];

}

main() {

int n = 14, k = 3;

JP5(n,k);

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### If you want...Ver 6

/\*

Name: JP.CPP (Ver. 6)

Copyright: (C) 2023

Author: Devcpp Fan

Date: 22-06-23 11:50

Description: Josephus Problem

'''

\*/

#include <bits/stdc++.h>

using namespace std;

/\*

vector<T> q;

q.push\_back(x); // nap x vao dau phai

x = q.front(); // xem dau trai

q.erase(q.begin()); // xoa dau trai

v.size();

v.clear();

v.back();

v.front();

v.empty();

\*/

#define Add Push

#define AddRight Push

#define GetLeft Get

#define Del Erase

#define Delete Erase

#define Front Get

class MyQueue {

public:

vector<int> Data;

MyQueue() { Data.clear(); }

inline void Push(int x) { // add x to the right

Data.push\_back(x);

}

inline Erase() { // delete elem in the left

Data.erase(Data.begin());

}

inline int TakeOut() { // takout lay phan tu tu dau trai

int x = Data.front(); // lay tri

Erase(); // xoa

}

inline int Get() { // xem dau trai

return Data.front();

}

inline void Move() { // lay trai -> nap phai

int x = Data.front();

Push(x);

Data.erase(Data.begin());

//Push(Take());

}

void Print(const char \* msg) {

cout << msg;

for (int i = 0; i < Data.size(); ++i) {

cout << " " << Data[i];

}

}

};

void Go() {

cout << " ? ";

fflush(stdin);

if (cin.get()=='.') exit(0);

}

void Print(int a[], int d, int c, const char \* msg = "") {

cout << msg;

for (int i = d; i <= c; ++i) {

cout << " " << a[i];

}

}

void Print(char b[], int d, int c, const char \* msg = "") {

cout << msg;

for (int i = d; i <= c; ++i) {

cout << " " << (int)b[i];

}

}

void Print(vector<int> a, const char \* msg = "") {

cout << msg;

for (int i = 0; i < a.size(); ++i) {

cout << " " << a[i];

}

}

// Using Queue

int JP4(int n, int k) {

MyQueue q;

for(int i = 1; i <= n; ++i) q.Push(i);

q.Print("\n Init: ");

for(int i = 1; i < n; ++i) { // lap n-1 lan

// chuyen k-1 phan tu dau ve cuoi

for(int j = 1; j < k; ++j) q.Move();

q.Erase(); // bo phan tu thu k

q.Print("\n q: "); //Go();

} // i

int res = q.Front();

cout << "\n Result: " << res;

return res;

}

main() {

int n = 14, k = 3;

JP4(n,k);

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### One more...Ver 7: Simplest queue

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *0* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* | *11* | *12* | *12* |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** |
| **L** |  |  |  |  |  |  |  |  |  |  |  |  | **R** |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |

/\*

Name: JP.CPP (Ver 7)

Copyright: (C) 2023

Author: Devcpp Fan

Date: 22-06-23 11:50

Description: Josephus Problem

simplest queue

'''

\*/

#include <bits/stdc++.h>

using namespace std;

int L, R, N;

int \*a;

void Go() {

cout << " ? ";

fflush(stdin);

if (cin.get()=='.') exit(0);

}

inline void Inc(int &i) { i = (i+1) % N; }

void Print(int a[], int d, int c, const char \* msg = "") {

cout << msg;

while(true) {

cout << " " << a[d];

if (d == c) return;

Inc(d);

}

}

int TakeOut() { // TakeOut the left

int x = a[L];

Inc(L);

return x;

}

void Push(int x) { // add x to the Rigjht

Inc(R);

a[R] = x;

}

void Move() { // Move L -> R

int x = TakeOut();

Push(x);

}

// Using simplest Queue

int JP7(int n, int k) {

N = n;

a = new int[N];

//fill(a, a+N, 0);

L = 0; R = N-1;

for(int i = 0; i < n; ++i) a[i] = i + 1;

Print(a, L, R, "\n Init: ");

for(int i = 1; i < N; ++i) { // loop n-1 times

// move k-1 elem front L to R

for(int j = 1; j < k; ++j) Move();

TakeOut(); // del

Print(a, L, R, "\n a: ");

}

int res = a[L];

cout << "\n Result: " << res;

return res;

}

main() {

int n = 14, k = 3;

JP7(n,k);

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### MyQueue Again

/\*

Name: JP.CPP (Ver 8)

Copyright: (C) 2023

Author: Devcpp Fan

Date: 22-06-23 11:50

Description: Josephus Problem

simplest queue

'''

\*/

#include <bits/stdc++.h>

using namespace std;

void Go() {

cout << " ? ";

fflush(stdin);

if (cin.get()=='.') exit(0);

}

class MyQueue {

public:

// Data

int \* Data;

int Len;

int L, R;

// Methods

inline MyQueue(int n) {

Len = n;

Data = new int[Len];

fill(Data, Data + Len, 0);

L = 0; R = Len-1;

}

inline Size() { return Len; }

inline void Inc(int &i) { i = (i+1) % Len; }

inline void Print(const char \* msg = "") {

int d = L, c = R;

cout << msg;

while(true) {

cout << " " << Data[d];

if (d == c) return;

Inc(d);

}

}

inline int TakeOut() { // TakeOut the left

int x = Data[L];

Inc(L);

return x;

}

inline void Push(int x) { // add x to the Rigjht

Inc(R);

Data[R] = x;

}

inline void Move() { // Move L -> R

Push(TakeOut());

}

inline int Left() {

return Data[L];

}

inline int Right() {

return Data[R];

}

};

// Using simplest Queue

int JP7(int n, int k) {

MyQueue q(n);

for(int i = 0; i < n; ++i)

q.Push(i+1);

q.Print("\n Init: "); Go();

for(int i = 1; i < n; ++i) { // loop n-1 times

// move k-1 elem front L to R

for(int j = 1; j < k; ++j) q.Move();

q.TakeOut(); // del

q.Print("\n q: ");

}

int res = q.Left();

cout << "\n Result: " << res;

return res;

}

main() {

int n = 14, k = 3;

JP7(n,k);

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### Using stringstream

/\*

Name: JP.CPP (Ver 9)

Copyright: (C) 2023

Author: Devcpp Fan

Date: 22-06-23 11:50

Description: Josephus Problem

stringstream

'''

\*/

#include <bits/stdc++.h>

using namespace std;

void Go() {

cout << " ? ";

fflush(stdin);

if (cin.get()=='.') exit(0);

}

// Using simplest Queue

int JP9(int n, int k) {

stringstream ss;

char BL = ' '; // 32

for(int i = 1; i <= n; ++i) {

ss << i; ss << BL;

}

int x;

for(int i = 1; i < n; ++i) { // loop n-1 times

for(int j = 1; j < k; ++j) {

ss >> x;

ss << BL << x;

}

ss >> x;

cout << "\n " << i << ": " << ss.str();

}

int res;

ss >> res;

cout << "\n Result: " << res;

return res;

}

main() {

int n = 14, k = 3;

JP9(n,k);

cout << endl << "\n\n T h e E n d \n";

return 0;

}

## **Rectangle Cipher**

String s, n = 3: con\*co\*bay\*la\*bay\*la

-> c\*\*yaalocb\*\*yanoalb\*

|  |  |  |
| --- | --- | --- |
| c | o | n |
| \* | c | o |
| \* | b | a |
| y | \* | l |
| a | \* | b |
| a | y | \* |
| l | a |  |

Application?

/\*

Name: RC.CPP

Copyright: (C) 2023

Author: Devcpp Fan

Date: 22-06-23 11:50

Description: Rectangle cipher

'''

\*/

#include <bits/stdc++.h>

using namespace std;

void Go() {

cout << " ? ";

fflush(stdin);

if (cin.get()=='.') exit(0);

}

string Encode(string s, int n) {

string w = "";

int len = s.length();

cout << "\n n = " << n << " len = " << len;

for (int j = 0; j < n; ++j) {

for (int i = j; i < len; i += n) {

w += s[i];

}

}

cout << "\n Encode: " << s << " -> " << w;

return w;

}

void Print(int a[], int d, int c, const char \*msg = "") {

cout << msg;

for(int i = d; i <= c; ++i) cout << " " << a[i];

}

string Decode(string w, int n) {

int len = w.length();

int m = len / n; // columns, n: rows

int r = len - m \* n; // du

cout << "\n n = " << n << " len = " << len;

int m1 = m;

if(r > 0) ++m1; // m1 = m+1

int d[n];

for(int i = 0; i < r; ++i) d[i] = m1;

for(int i = r; i < n; ++i) d[i] = m;

Print(d, 0, n-1, " \n d : ");

string s = "";

for(int j = 0; j < m1; ++j){

int i = j;

for(int k = 0; k < n; ++k) {

if(i < len) s += w[i];

if(s.length() == len) break;

i += d[k];

}

}

cout << "\n Decode: " << w << " -> " << s;

return s;

}

main() {

int n = 3;

string w = Encode("con\*co\*bay\*la\*bay\*la", n);

Decode(w, n);

string s = "The Duy Tan University";

int len = s.length();

for(n = 3; n < len; ++n) {

cout << "\n Codeing " << s << " for key = " << n;

w = Encode(s, n);

string v = Decode(w, n); Go();

}

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### A new Version

Adding an outside character

Encode(s)

Tim ki tu cmin khong co trong s

Them trai cmin de len(s) la boi cua n

Rai gianh the buoc n

Decode(w)

Tim ki tu cmin khong co trong w

len(w) luon luon la boi cua n (ko can them)

Rai gianh the buoc m

Algorithm is simple and more interesting...

/\*

Name: RC.CPP

Copyright: (C) 2023

Author: Devcpp Fan

Date: 22-06-23 11:50

Description: Rectangle cipher

Addind an ouside char

'''

\*/

#include <bits/stdc++.h>

using namespace std;

void Go() {

cout << " ? ";

fflush(stdin);

if (cin.get()=='.') exit(0);

}

char FindCmin(string s) {

char cmin = s[0]; // ' '

for(int i = 1; i < s.length(); ++i)

if (cmin > s[i]) cmin = s[i];

return cmin;

}

string Encode(string s, int n) {

cout << "\n Encoding... " ;

char cmin = FindCmin(s);

--cmin;

cout << "\n Input " << s << " cmin = " << cmin;

// them cho du boi so ua n

int len = s.length();

int r = n - len % n; // so ki tu can them

for(int i = 0; i < r; ++i) s += cmin;

len = s.length();

cout << "\n Now s = " << s << " len = " << len;

// len = boi so cua n

string w = "";

cout << "\n n = " << n << " len = " << len;

for (int j = 0; j < n; ++j) {

for (int i = j; i < len; i += n) {

w += s[i];

}

}

cout << "\n Encode: " << s << " -> " << w;

return w;

}

void Print(int a[], int d, int c, const char \*msg = "") {

cout << msg;

for(int i = d; i <= c; ++i) cout << " " << a[i];

}

string Decode(string w, int n) {

char cmin = FindCmin(w); // cmin ko co trong ban ro

cout << "\n Decoding... ";

cout << "\n Input " << w << " cmin = " << cmin;

// them cho du boi so ua n

int len = w.length(); // len = boi so cua n

len = w.length();

cout << "\n Now w = " << w << " len = " << len;

string s = "";

cout << "\n n = " << n << " len = " << len;

int m = len / n;

for (int j = 0; j < m; ++j) {

for (int i = j; i < len; i += m) {

if (w[i] != cmin) s += w[i];

}

}

cout << "\n Decode: " << w << " -> " << s;

return s;

}

main() {

int n = 3;

string w = Encode("con\*co\*bay\*la\*bay\*la", n);

Decode(w, n);

string s = "The Duy Tan University";

int len = s.length();

cout << "\n ----- LOOP ------";

for(n = 3; n < len; ++n) {

cout << "\n Codeing " << s << " for key = " << n;

w = Encode(s, n);

string v = Decode(w, n); Go();

}

cout << endl << "\n\n T h e E n d \n";

return 0;

}

## Farey Sequences

|  |
| --- |
| A silhouette of a person  Description automatically generated with medium confidence**John Farey Sr.**  1766 – 1826 |

*Nhà địa chất học người Anh John Farey (1766 − 1826) mô tả dãy Farey(n) là dãy các phân số tối giản được sắp tăng, có mẫu số không vượt quá giá trị n cho trước.*

Ví dụ

*Với n = 5 bạn có dãy Farey gồm 11 phân số như sau:*

*Bạn hãy viết dãy Farey tổng quát với giới hạn n là 100.*

Algorithm

Bạn có ba cách làm sau đây (xét với ví dụ n = 5):

### Version 1

Pha 1. Bạn tạo ra toàn bộ các phân số tối giản không trùng lặp với tử số *t* và mẫu số *m* biến thiên trong khoảng từ 1 đến *n*.

* Pha 2. Bạn sắp tăng các phân số:

Bạn biểu diễn mỗi phân số t/m dưới dạng danh sách hai phần tử:

[t,m]

Giả sử f là danh sách chứa kết quả khi thực hiện hàm Farey(n). Bạn khởi trị f với hai phân số nhỏ nhất 0/1 và lớn nhất của dãy 1/1:

f = [[0, 1], [1, 1]]

Bạn tạo phân số tối giản qua hàm sau đây:

Hàm này nhận vào hai giá trị nguyên dương t và m và cho ra phân số tối giản [t//d,m//d] với d là ước chung lớn nhất của t và m.

Ví dụ

Reduced(12,20) = [3,5] # d = Gcd(12,20) = 4

Mỗi khi tạo thêm một phân số mới x = [t,m] bạn kiểm tra, nếu x chưa có trong f thì bạn nạp vào f.

Để thực hiện Pha 2 ta gọi hàm sorted. Chú ý rằng có hai hàm sắp xếp dãy d là sort và sorted, cụ thể là

Ví dụ

[3, 5] : 0.6 # 3 / 5 = 0.6

[1, 2] : 0.5 # 1 / 2 = 0.5

Khi đó hàm sorted sẽ thay phép so sánh hai phân số [3,5] và [1,2] bằng phép so sánh hai số thực 0.6 và 0.5.

Program

/\*

Name: FS.CPP, Ver. 1

Copyright: (C) 2023

Author: Devcpp Fan

Date: 22-06-23 11:50

Description: Farey Sequence

'''

\*/

#include <bits/stdc++.h>

using namespace std;

int Gcd(int a, int b) {

return (b == 0) ? a : Gcd(b, a % b);

}

inline int Sign(int c) {

return (c == 0) ? 0

: ((c > 0) ? 1 : -1) ;

}

class Frac {

public :

int Num;

int Den;

// constructors

Frac() { Num = 0; Den = 1; }

Frac(int n, int d = 1) { Num = n; Den = d; }

Frac Reduce() {

int g = Gcd(abs(Num), Den);

Num /= g;

Den /= g;

return \*this;

}

void Print(const char \* msg = "") {

cout << msg;

cout << Num;

if(Num == 0 || Den == 1) return;

cout << "/" << Den;

}

};

typedef vector<Frac> VF;

void Print(VF a, const char \* msg = "") {

cout << msg;

for(int i = 0; i < a.size(); ++i)

a[i].Print(" ");

cout << "\n Total " << a.size();

}

inline int Compare(const Frac & x, const Frac & y) {

return Sign(x.Num\*y.Den - x.Den\*y.Num);

}

inline bool operator ==(const Frac & x, const Frac & y) {

return Compare(x,y) == 0;

}

bool Contains(VF a, Frac x) {

for(int i = 0; i < a.size(); ++i)

if (Compare(a[i], x) == 0)

return true;

return false;

}

inline bool Less(Frac x, Frac y) {

return x.Num\*y.Den < x.Den\*y.Num;

}

void Farey1(int n) {

VF a;

a.push\_back(Frac(0));

a.push\_back(Frac(1));

Print(a, "\n Init a:");

for(int d = 2; d <= n; ++d) {

for(int n = 1; n < d; ++n) {

Frac x(n,d);

if (Contains(a,x)) continue;

a.push\_back(x);

}

}

Print(a, "\n Init a:");

sort(a.begin(), a.end(), Less);

Print(a, "\n Sorted a:");

}

main() {

Farey1(5);

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### Version 2

Đầu tiên bạn làm quen với khái niệm *phân số trung bình* (*middle fraction*).

Đó chính là quy tắc "tử cộng tử, mẫu cộng mẫu".

Tiếp theo bạn thực hiện các bước sau đây:

* Bạn xuất phát từ hai phân số
* Bạn tạo ra phân số trung bình có mẫu số 2 và đặt vào giữa chúng:
* Bạn duyệt dãy trên để tạo ra các phân số trung bình mẫu số 3 và đặt vào giữa mỗi cặp phân số thành phần:
* Bạn thực hiện lại bước trên để tạo ra các phân số trung bình mẫu số 4, mẫu số 5 và đặt vào giữa mỗi cặp...

Trong chương trình dưới đây toán tử gán a = b tại dòng # (1) được thực hiện theo tham chiếu, nghĩa là a và b trỏ đến cùng một miền nhớ trong RAM.

Program

/\*

Name: FS.CPP, Ver. 2

Copyright: (C) 2023

Author: Devcpp Fan

Date: 22-06-23 11:50

Description: Farey Sequence

Median operator (%)

'''

\*/

#include <bits/stdc++.h>

using namespace std;

void Go() {

cout << " ? ";

fflush(stdin);

if (cin.get()=='.') exit(0);

}

int Gcd(int a, int b) {

return (b == 0) ? a : Gcd(b, a % b);

}

inline int Sign(int c) {

return (c == 0) ? 0

: ((c > 0) ? 1 : -1) ;

}

class Frac {

public :

int Num;

int Den;

// constructors

Frac() { Num = 0; Den = 1; }

Frac(int n, int d = 1) { Num = n; Den = d; }

Frac Reduce() {

int g = Gcd(abs(Num), Den);

Num /= g;

Den /= g;

return \*this;

}

void Print(const char \* msg = "") {

cout << msg;

cout << Num;

if(Num == 0 || Den == 1) return;

cout << "/" << Den;

}

};

typedef vector<Frac> VF;

void Print(VF a, const char \* msg = "") {

cout << msg;

for(int i = 0; i < a.size(); ++i)

a[i].Print(" ");

cout << "\n Total " << a.size();

}

inline int Compare(const Frac & x, const Frac & y) {

return Sign(x.Num\*y.Den - x.Den\*y.Num);

}

inline bool operator ==(const Frac & x, const Frac & y) {

return Compare(x,y) == 0;

}

// Median operator (PS trung binh)

inline Frac operator %(const Frac & x, const Frac & y) {

return Frac(x.Num+y.Num, x.Den+y.Den).Reduce();

}

bool Contains(VF a, Frac x) {

for(int i = 0; i < a.size(); ++i)

if (Compare(a[i], x) == 0)

return true;

return false;

}

inline bool Less(Frac x, Frac y) {

return x.Num\*y.Den < x.Den\*y.Num;

}

void Farey2(int n) {

VF a;

a.push\_back(Frac(0));

a.push\_back(Frac(1));

// a -> b

for(int den = 2; den <= n; ++den) {

// Print(a, "\n a: ");

VF b;

b.push\_back(a[0]);

for(int i = 1; i < a.size(); ++i) {

if(a[i-1].Den + a[i].Den == den)

b.push\_back(a[i-1] % a[i]);

b.push\_back(a[i]);

}

a = b; b.clear(); // Go();

}

Print(a, "\n Result a:");

}

main() {

Farey2(5);

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### Version 3

Ta sử dụng một số tính chất của dãy Farey để tiếp tục cải tiến thuật toán.

Nếu p1, p2 và p3 là ba phân số liên tiếp trong dãy Farey thì

(1) Tử(p2) \* Mẫu(p1) - Tử(p1) \* Mẫu(p2) = 1

(2) Mẫu(p1) + Mẫu(p2) > n

(3) p2 là phân số trung bình của p1 và p3

(4) Tử(p3) = v\*Tử(p2)−Tử(p1), Mẫu(p3) = v\*Mẫu(p2)−Mẫu(p1),

v = (Mẫu(p1)+n) div Mẫu(p2).

Từ tính chất 4 ta suy ra ngay cách xác định phân số thứ ba thông qua hai phân số sát trước.

Ví dụ

Với n = 5 ta khởi trị hai phân số đầu tiên của dãy Farey là:

p1 = 0/1, p2 = 1/5

Ta tính p3

v = (Mẫu(p1)+n) div Mẫu(p2) = (1+5) div 5 = 6 div 5 = 1

Tử(p3) = v \* Tử(p2) - Tử(p1) = 1\*1 - 0 = 1

Mẫu(p3) = v \* Mẫu(p2) - Mẫu(p1) = 1\*5 - 1 = 4

Vậy p3 = 1 / 4

Tiếp đến ta tính p4 theo p2 và p3,...

Trong chương trình dưới đây f[-1] là phần tử cuối của danh sách f chứa các phân số được sinh ra, f[-2] là phân số sát phân số cuối trong f.

Program

/\*

Name: FS.CPP, Ver. 3

Copyright: (C) 2023

Author: Devcpp Fan

Date: 22-06-23 11:50

Description: Farey Sequence

if a, b, and c are three consecutive fractions of a Farey sequence then:

(1) b.Num\*a.Den - a.Num\*b.Den = 1

(2) a.Den + b.Den > n

(3) b = a % c

(4) c.Num = v\*b.Num - a.Num, c.Den = v\*b.Den - a.Den

v = (a.Den + n) div b.Den.

'''

\*/

#include <bits/stdc++.h>

using namespace std;

void Go() {

cout << " ? ";

fflush(stdin);

if (cin.get()=='.') exit(0);

}

int Gcd(int a, int b) {

return (b == 0) ? a : Gcd(b, a % b);

}

inline int Sign(int c) {

return (c == 0) ? 0

: ((c > 0) ? 1 : -1) ;

}

class Frac {

public :

int Num;

int Den;

// constructors

Frac() { Num = 0; Den = 1; }

Frac(int n, int d = 1) { Num = n; Den = d; }

Frac Reduce() {

int g = Gcd(abs(Num), Den);

Num /= g;

Den /= g;

return \*this;

}

void Print(const char \* msg = "") {

cout << msg;

cout << Num;

if(Num == 0 || Den == 1) return;

cout << "/" << Den;

}

};

typedef vector<Frac> VF;

void Print(VF a, const char \* msg = "") {

cout << msg;

for(int i = 0; i < a.size(); ++i)

a[i].Print(" ");

cout << "\n Total " << a.size();

}

inline int Compare(const Frac & x, const Frac & y) {

return Sign(x.Num\*y.Den - x.Den\*y.Num);

}

inline bool operator ==(const Frac & x, const Frac & y) {

return Compare(x,y) == 0;

}

inline bool operator !=(const Frac & x, const Frac & y) {

return !(x == y);

}

// Median operator (PS trung binh)

inline Frac operator %(const Frac & x, const Frac & y) {

return Frac(x.Num+y.Num, x.Den+y.Den).Reduce();

}

bool Contains(VF a, Frac x) {

for(int i = 0; i < a.size(); ++i)

if (Compare(a[i], x) == 0)

return true;

return false;

}

inline bool Less(Frac x, Frac y) {

return x.Num\*y.Den < x.Den\*y.Num;

}

void Farey3(int n) {

VF res;

Frac a(0); // 0/1

Frac b(1,n); // 1/n

Frac c;

Frac one(1);

res.push\_back(a);

res.push\_back(b);

Print(res, "\n Init result: ");

while(c != one) {

// given a, b Find c ?

// v = (a.Den + n) / b.Den;

// c.Num = v\*b.Num - a.Num; c.Den = v\*b.Den - a.Den;

int v = (a.Den + n) / b.Den;

c.Num = v\*b.Num - a.Num;

c.Den = v\*b.Den - a.Den;

res.push\_back(c);

a = b; b = c;

}

Print(res, "\n Result: ");

}

main() {

Farey3(5);

cout << endl << "\n\n T h e E n d \n";

return 0;

}

## Roman Numerals

**Program**

/\*

Name: RN.CPP,

Copyright: (C) 2023

Author: Devcpp Fan

Date: 22-06-23 11:50

Description: Roman Numerals

\*/

#include <bits/stdc++.h>

using namespace std;

/\*----------------------------------

I V X L C D M (7)

I IV V IX X XL L XC C CD D CM M

4-9 Rules 2Char

XY

. X < Y: Y-X (4-9) IV XL

. X >= Y: X+Y III VI ...

----------------------------------\*/

// 1-1 mapping

const int INT[] =

{1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1};

const string ROME[] =

{ "M", "CM","D", "CD","C","XC","L","XL","X","IX","V","IV","I"};

const int MN = 13;

void Go() {

cout << " ? ";

fflush(stdin);

if (cin.get()=='.') exit(0);

}

// 124 -> "CXXIV"

// water Vase

string ToRome(int v) {

string s = "";

//while(v > 0) {

for(int i = 0; i < MN; ++i) {

while(v >= INT[i]) {

v -= INT[i];

s += ROME[i];

}

}

//}

return s;

}

// Find s n ROME

int Find(string s) {

for(int i = 0; i < MN; ++i)

if (s == ROME[i]) return i;

return -1;

}

/\*-----------------------------

two char ab

Find in ROME

if found at i return INT[i]

if Not: Find onla

MCDLXVII

MC ? false

M OK: 1000

CD OK: 1000 + 400 = 1400

LX false

L OK: 1400 + 50 = 1450

XV ? false

X OK : 1450 + 10 = 1560

VI ? false

V OK: 1450 + 5 = 1465

II ? false

I OK: 1465+1 = 1466

I OK: 1466 + 1 = 1467

--------------------------------\*/

int ToInt(string w) {

string r = "";

for(int i = 0; i < w.length(); ++i)

r += toupper(w[i]);

//cout << "\n " << r << " -> int ? ";

int res = 0;

int d = 1;

int len = r.length();

r += ".";

//cout << "\n Input: " << r;

for(int i = 0; i < len; i += d) {

string a = r.substr(i, 1);

string ab = r.substr(i, 2);

d = 1;

//cout << "\n ab = " << ab << " a = " << a;

int j = Find(ab);

// cout << "\n Find = " << j;

if(j >= 0) {

res += INT[j];

++d;

//cout << "\n ab = " << ab << " -> " << res << " d = " << d;

}

else {

res += INT[Find(a)];

//cout << "\n a = " << a << " -> " << res << " d = " << d;

}

//Go();

}

return res;

}

void Test1() {

for(int i = 1; i < 200; ++i)

cout << "\n " << i << " -> " << ToRome(i) ;

}

void Test2() {

int a[] = {9, 90, 49, 1999, 41, 2023, -1};

for(int i = 0; a[i] > 0; ++i)

cout << "\n " << a[i] << " -> " << ToRome(a[i]) ;

}

void Test3(int d, int c, bool wait = 0) {

int er = 0;

for(int i = d; i <= c; ++i) {

string s = ToRome(i);

int j = ToInt(s);

cout << "\n " << i << " -> " << s << " -> " << j;

if (i == j) cout << " CORRECT.";

else {

cout << " ERROR.";

++er;

}

if(wait) {

if (i % 20 == 0) Go();

}

}

cout << "\n Total ERROR = " << er;

}

main() {

// Test1();

// Test2();

// cout << "\n " << ToInt("MCDLxvii"); // 1467

Test3(1, 4000, 0);

cout << endl << "\n\n T h e E n d \n";

return 0;

}

## Expression

### Simple version

Compute the value of expression e, where variables are named a, b, .., z with the given values as

a : 0, b : 1, ..,z : 25.

Fot example

Val("(d+k)\*e-c\*d") = (3+10)\*4 - 2\*3 = 13\*4 - 6 = 52 - 6 = 46

**Algorithm**

Ta dùng kỹ thuật hai pha

* Pha 1. Dịch

Pha này có nhiệm vụ kiểm tra cú pháp của biểu thức đã cho. Nếu không phát hiện các lỗi cú pháp thì chuyển biểu thức thành dạng *Ba Lan* (ngược) và gọi Pha 2 để tính trị của biểu thức dạng Ba Lan. Ngược lại, nếu phát hiện dù chỉ một lỗi thì thông báo lỗi đó và không thực hiện Pha 2.

Ví dụ

Các lỗi cú pháp thường gặp có thể là

|  |  |  |
| --- | --- | --- |
| Biểu thức | Thông báo | Giải thích |
| (b+d)\*f-c) | Các cặp ngoặc không cân đối | Thiếu dấu mở ngoặc '(' |
| (b@d)\*f-c) | Sai dấu phép toán ? | @ không phải là phép toán |
| (b+X)\*(f-c) | Ký tự lạ ? | Biến X viết hoa |

Nếu không gặp lỗi cú pháp thì chương trình sẽ dịch biểu thức sang dạng sẵn sàng thực hiện. Trong trường hợp này ta chọn dạng *hậu tố* hay dạng *chuẩn Ba Lan*. Dạng cú pháp này do nhà toán học Ba Lan Jan Łukasiewicz đề xuất:

|  |
| --- |
| A person in a suit and tie  Description automatically generated with medium confidence |
| Jan Łukasiewicz  1878 − 1956 |

|  |  |
| --- | --- |
| Biểu thức | Dạng chuẩn Ba Lan |
| (b+d)\*(f-c) | bd+fc-\* |

Đặc điểm của ký pháp này là dấu phép toán đặt sau các toán hạng và do đó có thể tính giá trị của biểu thức thông qua một ngăn xếp.

* Pha 2. Tính trị

Tại pha này ta sẽ tính trị của biểu thức dạng Ba Lan thông qua một ngăn xếp (stack) st.

Ngăn xếp (stack) st là tổ chức dữ liệu theo nguyên tắc vào sau / ra trước: Giá trị nạp vào st sẽ được đặt tại vị trí cuối của st. Khi xuất một phần tử ra khỏi st thì giá trị này sẽ được lấy ra và ngăn xếp giảm đi một phần tử. Hình ảnh đơn giản nhất của ngăn xếp chính là một thùng xếp đồ có một đầu bịt kín.

Trong các ngôn ngữ lập trình, ngăn xếp thường được cài đặt đơn giản bằng một mảng (danh sách) một chiều.

Trong hình minh họa, ngăn xếp st được đặt nằm ngang với đầu vào/ra ở bên phải, đáy ở bên trái.

Lúc đầu bạn khởi trị ngăn xếp rỗng

st = []

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  | 🡨 🡪 đầu vào/ra |  |
| *Đáy* | | | | | | | |  |

Muốn nạp phần tử x vào ngăn xếp bạn viết

st.append(x)

Giả sử bạn đã nạp vào ngăn xếp st các phần tử 5, 7, 8, 2, 4, 3, 9. Nay bạn muốn nạp phần tử 6 vào st (Hình a):

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | 7 | 8 | 2 | 4 | 3 | 9 |  | **←** 6 | |
| *Đáy* | *a) Cần nạp trị 6 Ngọn* | | | | | | | |

Bạn viết

st.append(6)

Sau khi nạp 6 vào st bạn thu được (Hình b):

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | 7 | 8 | 2 | 4 | 3 | 9 | 6 |  |
| *Đáy* | *b) Sau khi nạp 6* | | | | | | *Ngọn* |  |

Muốn lấy một phần tử ra khỏi st ta viết:

y = st.pop()

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | 7 | 8 | 2 | 4 | 3 | 9 | 6 | → | |
| *Đáy* | *c) Lấy ngọn ngăn xếp* | | | | | | *Ngọn* |  |

Đến đây, bạn thu được

y = 6

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | 7 | 8 | 2 | 4 | 3 | 9 |  |  |
| *Đáy* | *c) Lấy ngọn ngăn xếp* | | | | | *Ngọn* |  |  |

và ngăn xếp giảm đi một phần tử.

Muốn xem giá trị hiện có trên ngọn của ngăn xếp st (nhưng không lấy ra), bạn viết:

st[-1]

trong đó, như bạn đã biết st[-1] truy cập đến phần tử cuối của danh sách.

Muốn kiểm tra ngăn xếp st có rỗng hay không bạn viết

if len(st) == 0:

Tóm tắt các thao tác trên ngăn xếp st

|  |  |
| --- | --- |
| Cú pháp | Ngữ nghĩa |
| st = [] | *Khởi trị st rỗng* |
| st.append(x) | *Nạp x vào st* |
| y = st.pop() | *Lấy ngọn ra khỏi st* |
| st[-1] | *Xem ngọn st* |
| len(st) == 0 | *Kiểm tra st rỗng ?* |
| *Các thao tác trên ngăn xếp st* | |

Giả sử ta đã hoàn tất Pha 1 và không gặp lỗi cú pháp, ta muốn thực hiện Pha 2 để tính trị của biểu thức dạng Ba Lan ghi trong string pol.

Pha 2. Hàm Exe(pol)

Hàm này tính trị cho biểu thức dạng Ba Lan, ghi trong biến string pol.

Hàm khởi tạo một ngăn xếp st rỗng

sau đó đọc từng ký tự c của dạng Ba Lan pol

và quyết định:

* Nếu là biến a,...z: Nạp trị của biến vào st
* Nếu là phép toán hai ngôi như a b +, a b -, a b \*, a b /: Thực hiện phép toán với hai trị trên cùng của st và ghi kết quả trung gian vào ngăn dưới. Giả sử c là phép cộng:

**Program**

/\*

Name: Expression.CPP (Ver 1)

Copyright: (C) 2023

Author: Devcpp Fan

Date: 22-06-23 11:50

Description:

The value of expression e, where variables are named

a, b, .., z with the given values as

a : 0, b : 1, ..,z : 25.

Example

Val("(d+k)\*e-c\*d") = (3+10)\*4 - 2\*3 = 13\*4 - 6 = 52 - 6 = 46

\*/

#include <bits/stdc++.h>

using namespace std;

const string OPE = "+-\*/%";

void Go() {

cout << " ? ";

fflush(stdin);

if (cin.get()=='.') exit(0);

}

bool Var(char c) { return (c >= 'a' && c <= 'z'); }

bool Ope(char c) {

for(int i = 0; i < OPE.length(); ++i)

if(OPE[i] == c) return 1;

return 0;

}

int Deg(char c) {

switch(c) {

case '\*': //return 100;

case '/': //return 100;

case '%': return 100;

case '+': // return 200;

case '-': return 200;

} // switch

return 1000;

}

// Phase 1: e -> pol(Lukasiewicz Polish Form)

// e = "(d+k)\*e-c\*d";

// pol = "dk+e\*cd\*-"

string Compiler(string e) {

string pol = "";

int len = e.length();

char st[len];

int p = -1;

st[++p] = '@';

for(int i = 0; i < len; ++i) {

char c = e[i];

if (Var(c)) { //

pol += c; // -> pol

continue;

}

if (Ope(c)) {

// st -> pol if st < c

while(Deg(st[p]) <= Deg(c)) {

pol += st[p--]; // -> st

}

st[++p] = c;

continue;

}

if (c == '(') {

st[++p] = c; // -> st

continue;

}

if (c == ')') {

// st -> pol

while(st[p] != '(') {

pol += st[p--];

}

p--; // bo (

continue;

}

}

// therest symbol in st

while(st[p] != '@') pol += st[p--];

return pol;

}

int Exe(string pol) {

cout << "\n Exe " << pol;

int len = pol.length();

int st[len];

int p = -1;

for(int i = 0; i < len; ++i) {

if (Var(pol[i])) {

st[++p] = pol[i] - 'a';

continue;

}

switch(pol[i]) {

case '+': st[p-1] += st[p]; --p; break;

case '-': st[p-1] -= st[p]; --p; break;

case '\*': st[p-1] \*= st[p]; --p; break;

case '/': st[p-1] /= st[p]; --p; break;

case '%': st[p-1] %= st[p]; --p; break;

} // switch

} // for

return st[0];

}

int Val(string e) {

string pol = Compiler(e);

cout << "\n " << e << " -> " << pol;

return Exe(pol);

}

main() {

string e = "(d+k)\*e-c\*d";

cout << "\n " << Val(e); // 46

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### Expression with Numbers

/\*

Name: Expression.CPP (Ver 2)

Copyright: (C) 2023

Author: Devcpp Fan

Date: 22-06-23 11:50

Description:

The value of expression e, where variables are named

a, b, .., z with the given values as

a : 0, b : 1, ..,z : 25

and int num and Rome Num

Example

Val("(d+k)\*IV-c\*3") = (3+10)\*4 - 2\*3 = 13\*4 - 6 = 52 - 6 = 46

\*/

#include <bits/stdc++.h>

using namespace std;

const string OPE = "+-\*/%><!";

const int MN = 13;

const int INT[] =

{1000,900,500,400,100, 90, 50, 40, 10, 9, 5, 4, 1};

const string ROME[] =

{"M","CM","D","CD","C","XC","L","XL","X","IX","V","IV","I"};

const string ROMESYM = "MDCLXVI";

typedef vector<int> VI;

typedef long long Long;

VI qconst; // const val

void Go() {

cout << " ? ";

fflush(stdin);

if (cin.get()=='.') exit(0);

}

bool Var(char c) { return (c >= 'a' && c <= 'z'); }

bool Ope(char c) {

for(int i = 0; i < OPE.length(); ++i)

if(OPE[i] == c) return 1;

return 0;

}

bool Num(char c) { return (c >= '0' && c <= '9'); }

bool Rome(char c) {

for(int i = 0; i < ROMESYM.length(); ++i)

if(ROMESYM[i] == c) return 1;

return 0;

}

int Deg(char c) {

switch(c) {

case '!':

case '>':

case '<': return 100;

case '\*':

case '/':

case '%': return 200;

case '+':

case '-': return 300;

} // switch

return 1000;

}

int GetInt(string s) {

stringstream ss;

ss << s;

int x;

ss >> x;

return x;

}

// search s in ROME

int Find(string s) {

for(int i = 0; i < MN; ++i)

if (ROME[i] == s) return i;

return -1;

}

int GetRome(string s) {

int len = s.length();

s += "#";

int val = 0, d = 1;

for(int i = 0; i < len; i += d) {

string a = s.substr(i,1);

string ab = s.substr(i,2);

d = 1;

int j = Find(ab);

if(j >= 0) {

val += INT[j];

++d;

} else val += INT[Find(a)];

}

return val;

}

// Phase 1: e -> pol(Lukasiewicz Polish Form)

// e = "(d+k)\*e-c\*d";

// pol = "dk+e\*cd\*-"

string Compiler(string e) {

string pol = "";

int len = e.length();

char st[len];

int p = -1;

st[++p] = '@';

qconst.clear();

for(int i = 0; i < len; ++i) {

char c = e[i];

if (Num(c)) {

int j = i+1;

while(Num(e[j])) ++j;

qconst.push\_back(GetInt(e.substr(i, j-i)));

pol += "#";

i = j-1;

continue;

}

if (Rome(c)) {

int j = i+1;

while(Rome(e[j])) ++j;

qconst.push\_back(GetRome(e.substr(i, j-i)));

pol += "#";

i = j-1;

continue;

}

if (Var(c)) { //

pol += c; // -> pol

continue;

}

if (Ope(c)) {

// st -> pol if st < c

while(Deg(st[p]) <= Deg(c)) {

pol += st[p--]; // -> st

}

st[++p] = c;

continue;

}

if (c == '(') {

st[++p] = c; // -> st

continue;

}

if (c == ')') {

// st -> pol

while(st[p] != '(') {

pol += st[p--];

}

p--; // bo (

continue;

}

}

// therest symbol in st

while(st[p] != '@') pol += st[p--];

return pol;

}

Long Fac(Long n) {

return (n <= 1) ? 1 : n\*Fac(n-1);

}

Long Exe(string pol) {

cout << "\n Exe " << pol;

int len = pol.length();

Long st[len];

int p = -1;

int q = -1; // for val

for(int i = 0; i < len; ++i) {

char c = pol[i];

if (Var(c)) {

st[++p] = pol[i] - 'a';

continue;

}

if (c == '#') {

st[++p] = qconst[++q];

continue;

}

switch(c) {

case '!': st[p] = Fac(st[p]); break;

case '>': ++st[p]; break;

case '<': --st[p]; break;

case '+': st[p-1] += st[p]; --p; break;

case '-': st[p-1] -= st[p]; --p; break;

case '\*': st[p-1] \*= st[p]; --p; break;

case '/': st[p-1] /= st[p]; --p; break;

case '%': st[p-1] %= st[p]; --p; break;

} // switch

} // for

return st[0];

}

int Val(string e) {

string pol = Compiler(e);

cout << "\n " << e << " -> " << pol;

return Exe(pol);

}

main() {

string e = " (( d + k)\*IV - (b>)\*(5<<) -(XL + II \* 3) ) >> \* (16 - VI) ";

cout << "\n " << Val(e); // 46

// cout << "\n " << Fac(6);

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### The next steps

**Error capture**

Syntax error

* e = (x + y)): expected (
* e = ((x + y): expected )
* e = x ? y:: unexpected symbol ?
* e = ab: syntax error
* ...

Run error

* 2 / 0, 2 % 0: Duvide by zero
* -2 !: negative parameter
* ...

**New operations**

* x & y: greatest common divisor (uoc chung lon nhat)
* x | y: lowest common multiple (boi chung nho nhat)
* x ^ y: power
* ~: negative operation
* F n: get n-th Fibonacci number: F(0) = F(1) = 1; F(n) = F(n-2)+F(n-1), n >= 2
* ...

Program

/\*

Name: Expression.CPP (Ver 3)

Copyright: (C) 2023

Author: Devcpp Fan

Date: 22-06-23 11:50

Description:

The value of expression e, where variables are named

a, b, .., z with the given values as

a : 0, b : 1, ..,z : 25

and int num and Rome Num

Example

Val("(d+k)\*IV-c\*3") = (3+10)\*4 - 2\*3 = 13\*4 - 6 = 52 - 6 = 46

x + y, x - y, x \* y, x / y. x % y: Arithmetic operations

x & y: greatest common divisor (uoc chung lon nhat)

x | y: lowest common multiple (boi chung nho nhat)

x ^ y: power

~: negative operation

F: generate n-th Fibonacci number

\*/

#include <bits/stdc++.h>

using namespace std;

const char BL = ' '; // 32

const string OPE = "+-\*/%><!&|^~F";

const int MN = 13;

const int INT[] =

{1000,900,500,400,100, 90, 50, 40, 10, 9, 5, 4, 1};

const string ROME[] =

{"M","CM","D","CD","C","XC","L","XL","X","IX","V","IV","I"};

const string ROMESYM = "MDCLXVI";

typedef vector<int> VI;

typedef long long Long;

VI qconst; // const val

void Go() {

cout << " ? ";

fflush(stdin);

if (cin.get()=='.') exit(0);

}

void SynErr(int e, char c = BL) {

cout << "\n Syntax Error " << e << ": " << c;

switch(e) {

case 1: cout << " expected )"; break;

case 2: cout << " expected ("; break;

case 3: cout << " unexpected symbol" ; break;

case 4: cout << "?"; break;

}

exit(e);

}

void RunErr(int e, Long v) {

cout << "\n Running Error " << e << ": " << v;

switch(e) {

case 1: cout << " divide by zero"; break;

case 2: cout << " negative parameter"; break;

}

exit(e);

}

bool Var(char c) { return (c >= 'a' && c <= 'z'); }

bool Ope(char c) {

for(int i = 0; i < OPE.length(); ++i)

if(OPE[i] == c) return 1;

return 0;

}

bool Num(char c) { return (c >= '0' && c <= '9'); }

bool Rome(char c) {

for(int i = 0; i < ROMESYM.length(); ++i)

if(ROMESYM[i] == c) return 1;

return 0;

}

int Deg(char c) {

switch(c) {

case 'F':

case '~':

case '!':

case '>':

case '<': return 100;

case '&':

case '|':

case '^':

case '\*':

case '/':

case '%': return 200;

case '+':

case '-': return 300;

} // switch

return 1000;

}

int GetInt(string s) {

stringstream ss;

ss << s;

int x;

ss >> x;

return x;

}

// search s in ROME

int Find(string s) {

for(int i = 0; i < MN; ++i)

if (ROME[i] == s) return i;

return -1;

}

int GetRome(string s) {

int len = s.length();

s += "#";

int val = 0, d = 1;

for(int i = 0; i < len; i += d) {

string a = s.substr(i,1);

string ab = s.substr(i,2);

d = 1;

int j = Find(ab);

if(j >= 0) {

val += INT[j];

++d;

} else val += INT[Find(a)];

}

return val;

}

// Phase 1: e -> pol(Lukasiewicz Polish Form)

// e = "(d+k)\*e-c\*d";

// pol = "dk+e\*cd\*-"

string Compiler(string s) {

string e = "";

for(int i = 0; i < s.length(); ++i)

if (s[i] != BL) e += s[i]; // space symbol 32

cout << "\n " << e;

string pol = "";

int len = e.length();

char st[len];

int p = -1;

st[++p] = '@';

qconst.clear();

for(int i = 0; i < len; ++i) {

char c = e[i];

if (Num(c)) {

int j = i+1;

while(Num(e[j])) ++j;

qconst.push\_back(GetInt(e.substr(i, j-i)));

pol += "#";

i = j-1;

continue;

}

if (Rome(c)) {

int j = i+1;

while(Rome(e[j])) ++j;

qconst.push\_back(GetRome(e.substr(i, j-i)));

pol += "#";

i = j-1;

continue;

}

if (Var(c)) { //

pol += c; // -> pol

continue;

}

if (Ope(c)) {

if(c == '+') {

if(i == 0 || e[i-1] == '(')

continue;

}

if(c == '-') {

if(i == 0 || e[i-1] == '(')

c = '~';

}

// st -> pol if st < c

while(Deg(st[p]) <= Deg(c)) {

pol += st[p--]; // -> st

}

st[++p] = c; // c -> st

continue;

}

if (c == '(') {

st[++p] = c; // c -> st

continue;

}

if (c == ')') { // search symbol (

// st -> pol

while(st[p] != '@') {

if (st[p] == '(') break;

pol += st[p--];

}

if (st[p] != '(') SynErr(2);

p--; // bo (

continue;

}

SynErr(3, c);

}

// therest symbol in st

while(st[p] != '@') {

if (!Ope(st[p])) SynErr(3, st[p]);

pol += st[p--];

}

return pol;

}

Long Fac(Long n) {

return (n <= 1) ? 1 : n\*Fac(n-1);

}

Long Gcd(Long a, Long b) {

a = abs(a); b = abs(b);

return (b == 0) ? a : Gcd(b, a % b);

}

Long Lcm(Long a, Long b) {

return a\*(b / Gcd(a,b));

}

// F(0) = F(1) = 1

// F(n) = F(n-2) + F(n-1)

Long Fib(int n) {

if (n < 2) return 1;

Long a = 1 , b = 1, c;

for(int i = 2; i <= n; ++i) {

c = a + b;

a = b; b = c;

}

return c;

}

Long Explain(string pol) {

cout << "\n Explain: ";

int len = pol.length();

Long st[len];

int p = -1;

int q = -1; // for val

for(int i = 0; i < len; ++i) {

char c = pol[i];

if (Var(c)) {

cout << (c - 'a') << BL;

continue;

}

if (c == '#') {

cout << qconst[++q] << BL;

continue;

}

cout << c << BL;

}

}

Long Exe(string pol) {

cout << "\n Exe " << pol;

Explain(pol);

int len = pol.length();

Long st[len];

int p = -1;

int q = -1; // for val

for(int i = 0; i < len; ++i) {

char c = pol[i];

if (Var(c)) {

st[++p] = pol[i] - 'a';

continue;

}

if (c == '#') {

st[++p] = qconst[++q];

continue;

}

switch(c) {

case 'F': if (st[p] < 0) RunErr(2, st[p]);

st[p] = Fib(st[p]); break;

case '~': st[p] = -st[p]; break;

case '!': if (st[p] < 0) RunErr(2, st[p]);

st[p] = Fac(st[p]); break;

case '>': ++st[p]; break;

case '<': --st[p]; break;

case '+': st[p-1] += st[p]; --p; break;

case '-': st[p-1] -= st[p]; --p; break;

case '\*': st[p-1] \*= st[p]; --p; break;

case '/': if(st[p] == 0) RunErr(1, st[p]);

st[p-1] /= st[p]; --p; break;

case '%': if(st[p] == 0) RunErr(1, st[p]);

st[p-1] %= st[p]; --p; break;

case '&': st[p-1] = Gcd(st[p-1], st[p]); --p; break;

case '|': st[p-1] = Lcm(st[p-1], st[p]); --p; break;

case '^': st[p-1] = (Long)pow(st[p-1], st[p]); --p; break;

} // switch

} // for

return st[0];

}

int Val(string e) {

string pol = Compiler(e);

cout << "\n " << e << " -> " << pol;

return Exe(pol);

}

main() {

string e1 = "-c<\*5>>"; // ((-2)-1) \* (5+2) = -3\*7 = -21

cout << "\n " << Val(e1); // -21

string e2 = " (d + k) \* IV"; // (3+10)\*4 = 13\*4 = 52

cout << "\n " << Val(e2); // 52

string e = e1 + " - " + e2;

cout << "\n " << Val(e); // -21 - 52 = -73

cout << endl << "\n\n T h e E n d \n";

return 0;

}

## Stack & Queue

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

## Back Tracking

Diagram

Description automatically generated

29 June 2023