

Experiment No - 1
Problem A) The $3n + 1$ Problem

Source code:

```
import java.util.*;
public class OneA {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter value for n : ");
        int n = sc.nextInt();
        while (n >= 1) {
            System.out.print(n + " ");
            if (n % 2 == 0) {
                n = n / 2;
            } else {
                n = n * 3 + 1;
            }
            if(n == 1){
                break;
            }
        } System.out.print(n);
    }
}
```

Output:

Output Clear

```
java -cp /tmp/xFNG0BcNKK OneA
Enter value for n : 22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
```

Experiment No - 1

Problem B) The Trip

Source code:

```
import java.util.*;
public class OneA {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of friends: ");
        int n = sc.nextInt();
        float[] a = new float[n];
        for(int i = 0; i < n; i++){
            a[i]=sc.nextInt();
        }
        Arrays.sort(a);
        float mid = a[(n/2)];
        float start = a[0];
        System.out.println("Money must be exchanged to equalize the students costs : 
"+(mid-start));
    }
}
```

Output:

Output Clear

```
java -cp /tmp/PtVEf5xvyB OneA
Enter number of friends: 3
10
20
30
Money must be exchanged to equalize the students costs : 10.0
```

Experiment No. : 2

A) LCD Display

PROGRAM :

```
#include<stdio.h>
char type[5][6] = {
    {" "}, {" |"}, {" |"}, {" |"}, {" |"}, {" -- "},
};
char Num[10][5] = {
    {4,3,0,3,4}, /*0*/
    {0,1,0,1,0}, /*1*/
    {4,1,4,2,4}, /*2*/
    {4,1,4,1,4}, /*3*/
    {0,3,4,1,0}, /*4*/
    {4,2,4,1,4}, /*5*/
    {4,2,4,3,4}, /*6*/
    {4,1,0,1,0}, /*7*/
    {4,3,4,3,4}, /*8*/
    {4,3,4,1,4}, /*9*/
};
void PrintNum(char s[], int n)
{int a, b, c, tn;
for(a = 0; a < 5; a++) {
    if(a == 0 || a == 2 || a == 4) tn = 1;
    else tn = n;
    while(tn--) {
        for(b = 0; s[b]; b++) {
            if(s[b] == ' ') {putchar(' ');continue;}
            putchar(type[Num[s[b] - '0']][a][0]);
            for(c = 0; c < n; c++)
                putchar(type[Num[s[b] - '0']][a][1]);
            putchar(type[Num[s[b] - '0']][a][3]);
            if(s[b+1] != '\0') putchar(' ');
        }
        puts("");
    }
}
}
main()
{int n;
char s[20];
while(scanf("%d %s", &n, s) == 2)
{if(n == 0)
    break;
    PrintNum(s, n);
    puts("");
}
return 0;
}
```

OUTPUT:

```
2 1234
      --  --
    |  |  |  |
    |  |  |  |
      --  --  --
    | |  |  |
    | |  |  |
      --  --
```

Experiment No: 2

B)Graphical Editor

Program:

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <stdbool.h>

char Order;

char P[ 250 ][ 250 ];

int X, Y;

void _I( int _M, int _N )

{

    memset( P, 'O', sizeof( P ) );

    X = _M;

    Y = _N;

    return;

}

void _C( )

{

    memset( P, 'O' , sizeof( P ) );

    return;

}

void _L( int _X, int _Y, char Color )

{

    P[ _Y - 1 ][ _X - 1 ] = Color;

    return;

}

void _V( int _X, int _Y1, int _Y2, char Color )

{

    int temp;
```

```

    if( _Y1 > _Y2 )
        {temp = _Y1;
          _Y1 = _Y2;
          _Y2 = temp;
        }

    int i;

    _X --;

    for( i = _Y1 - 1; i < _Y2; i ++ ) P[ i ][ _X ] = Color;

    return;

}

void _H( int _X1, int _X2, int _Y, char Color )
{
    int temp;

    if( _X1 > _X2 )
        {temp = _X1;
          _X1 = _X2;
          _X2 = temp;
        }

    int i;

    _Y --;

    for( i = _X1 - 1; i < _X2; i ++ ) P[ _Y ][ i ] = Color;

    return;

}

void _K( int _X1, int _Y1, int _X2, int _Y2, char Color )
{
    int temp;

    if( _Y1 > _Y2 )
        {temp = _Y1;
          _Y1 = _Y2;
          _Y2 = temp;
        }

    if( _X1 > _X2 )

```

```

    {temp = _X1;

    _X1 = _X2;
    _X2 = temp;

}

int i, j;

for( j = _Y1 - 1; j < _Y2; j ++ )

    { for( i = _X1 - 1; i < _X2; i ++ )

        {

            P[ j ][ i ] = Color;

        }

    }

return;

}

void _F_Helper( int _X, int _Y, char DefaultColor, char AimColor )

{

    if( P[ _Y ][ _X ] == DefaultColor ) P[ _Y ][ _X ] = AimColor;//Here _Y, _X don't need minus 1

    else return;


    if( _X - 1 >= 0 ) _F_Helper( _X - 1, _Y, DefaultColor, AimColor );

    if( _X + 1 < X ) _F_Helper( _X + 1, _Y, DefaultColor, AimColor );

    if( _Y - 1 >= 0 ) _F_Helper( _X, _Y - 1, DefaultColor, AimColor );

    if( _Y + 1 < Y ) _F_Helper( _X, _Y + 1, DefaultColor, AimColor );

    return;

}

void _F( int _X, int _Y, char Color )

{

    _F_Helper( _X - 1, _Y - 1, P[ _Y - 1 ][ _X - 1 ], Color );

    return;

}

void _S( char *N )

{

```

```

int i, j;

printf( "%s\n", N );

for( i = 0; i < Y; i ++ ) {
    for( j = 0; j < X; j ++ )

        { printf( "%c",

            P[ i ][ j ] );

        }

    printf( "\n" );
}

return;
}

int main( )
{
    char COM[ 2 ];
    int INT[ 5 ];
    char Name[ 100 ];
    while( scanf( "%c", &Order ), Order != 'X' )

        {if( Order == 'I' ) {

            scanf( "%d%d", &INT[ 0 ], &INT[ 1 ] );

            _I( INT[ 0 ], INT[ 1 ] );

        }

        else if( Order == 'C' ) {

            _C( );

        }

        else if( Order == 'L' ) {

            scanf( "%d%d%s", &INT[ 0 ], &INT[ 1 ], COM );

            _L( INT[ 0 ], INT[ 1 ], COM[ 0 ] );

        }

        else if( Order == 'V' ) {

            scanf( "%d%d%d%s", &INT[ 0 ], &INT[ 1 ], &INT[ 2 ], COM );

            _V( INT[ 0 ], INT[ 1 ], INT[ 2 ], COM[ 0 ] );

        }
}

```



```

else if( Order == 'H' ) {

    scanf( "%d%d%d%s", &INT[ 0 ], &INT[ 1 ], &INT[ 2 ], COM );

    _H( INT[ 0 ], INT[ 1 ], INT[ 2 ], COM[ 0 ] );

}
else if( Order == 'K' ) {

    scanf( "%d%d%d%d%s", &INT[ 0 ], &INT[ 1 ], &INT[ 2 ], &INT[ 3 ], COM );

    _K( INT[ 0 ], INT[ 1 ], INT[ 2 ], INT[ 3 ], COM[ 0 ] );

}
else if( Order == 'F' ) {

    scanf( "%d%d%s", &INT[ 0 ], &INT[ 1 ], COM );

    _F( INT[ 0 ], INT[ 1 ], COM[ 0 ] );

}
else if( Order == 'S' )

    { scanf( "%s",

        Name );

        _S( Name );

    }

else if( Order != '\n' ) {

    gets( Name );/*As an empty line*/

}

else { }

}

return 0;

}

```

OUTPUT:

```
I 5 6
L 2 3 A

G 2 3 J
F 3 3 J
V 2 3 4 W
H 3 4 2 Z
```

```
S one.bmp
one.bmp
nnnnn
nnZZn
nWnnn
nWnnn
nnnnn
nnnnn
```

```
S two.bmp
two.bmp
nnnnn
nnZZn
nWnnn
nWnnn
nnnnn
nnnnn
```


Experiment 3
A) Interpreter

Program:

```
#include <iostream>
#include <stdio.h>
#include <sstream>
#include <iomanip>
#include <stdlib.h>
using namespace std;
int ram[1001][3];
int reg[10][3];
int getRam(int pos){
    return ram[pos][0]*100+ram[pos][1]*10+ram[pos][2];
}
int getReg(int pos){
    return reg[pos][0]*100+reg[pos][1]*10+reg[pos][2];
}
void setRam(int value, int
    pos){int v0 = value/100;
    int v1 = (value/10)%10;
    int v2 = value%10;
    ram[pos][0] = v0;
    ram[pos][1] = v1;
    ram[pos][2] = v2;
}
void setReg(int value, int
    pos){int v0 = value/100;
    int v1 = (value/10)%10;
    int v2 = value%10;
    reg[pos][0] = v0;
    reg[pos][1] = v1;
    reg[pos][2] = v2;
}
int main()
{
    int ncases, nwords, instructions, pc, param1, param2, command;
    bool flag, end;
    string aux;
    getline(cin, aux);
    ncases = atoi(aux.c_str());
    getline(cin, aux);
    while(ncases){
        flag = true;
        nwords = 0;
        for(int i = 0; i < 10; i++)
            for(int j = 0; j < 3; j++)
                reg[i][j] = 0;
        while(flag&&getline(cin,
            aux)){stringstream ss;
            if(aux.compare("")!=0){
                ss << setfill('0') << setw(3) << aux;

                ram[nwords][0] = ss.str()[0]-'0';
                ram[nwords][1] = ss.str()[1]-'0';
                ram[nwords][2] = ss.str()[2]-'0';
                nwords++;
            }
        }
    }
}
```

```

        else
            flag = false;
    }
    end = false;
    pc = 0;
    instructions = 0;
    while(!end){
        command = ram[pc][0];
        param1 = ram[pc][1];
        param2 = ram[pc][2];
        instructions++;
        switch(command){
            case 1:
                end = true;
                break;
            case 2:
                setReg(param2, param1);
                pc++;
                break;
            case 3:
                setReg((getReg(param1)+param2)%1000, param1);
                pc++;
                break;
            case 4:
                setReg((getReg(param1)*param2)%1000, param1);
                pc++;
                break;
            case 5:
                setReg(getReg(param2), param1);
                pc++;
                break;
            case 6:
                setReg((getReg(param1)+getReg(param2))%1000, param1);
                pc++;
                break;
            case 7:
                setReg((getReg(param1)*getReg(param2))%1000, param1);
                pc++;
                break;
            case 8:
                setReg(getRam(getReg(param2)),param1);
                pc++;
                break;
            case 9:
                setRam(getReg(param1),getReg(param2));
                pc++;
                break;
            case 0:
                if(getReg(param2)==0) pc++;
                else pc = getReg(param1);
                break;
        }
    }
    ncases--;
    cout << instructions << endl;
    if(ncases > 0) cout << endl;
}
return 0;

```

```
}
```

Output

```
1
```

```
299
```

```
492
```

```
495
```

```
399
```

```
492
```

```
495
```

```
399
```

```
283
```

```
279
```

```
689
```

```
078
```

```
100
```

```
000
```

```
000
```

```
000
```

```
16
```

Experiment 3

B) Austrailian voting

Program:

```
#include <iostream>
#include <string>
#include <sstream>

void parse_case(int &candidates_count, int &votes_count, char candidates[][80], int votes[][1000])
{
    std::cin >> candidates_count;
    std::cin.get();
    for (int j=0; j<candidates_count; j++)
    {
        std::cin.getline(candidates[j], sizeof(candidates[j]), '\n');
    }
    votes_count = 0;
    std::string line;
    while (std::getline(std::cin, line)) {
        if (line.empty())
            break;
        std::stringstream ss(line);
        for (int k=0; k<candidates_count; k++)
            {ss >> votes[k][votes_count];
        }
        votes_count++;
    }
}

void reset_losers(bool losers[20])
{
    for (int k=0; k<20; k++) {
        losers[k] = false;
    }
}

void count_votes(int candidates_count, int votes_count, int candidate_votes[20], int votes[][1000])
{
    for (int j=0; j<candidates_count; j++) {
        candidate_votes[j] = 0;
    }
    for (int j=0; j<votes_count; j++)
    {
        candidate_votes[(votes[0][j]-1)]++;
    }
}

int find_winners(int candidates_count, int remaining_candidates_count, int votes_count, int candidate_votes[20], int
subroutine_results[20]) {
    int max_votes = 0;
    int winners_count = 0;

    for (int k=0; k<candidates_count; k++)
    {
        if (candidate_votes[k] > max_votes)
        {
            max_votes = candidate_votes[k];
            subroutine_results[0] = k;
            winners_count = 1;
        }
        else if (candidate_votes[k] == max_votes)
        {
            subroutine_results[++winners_count-1] = k;
        }
    }
}
```

```

    }
}

if (winners_count == remaining_candidates_count || ( winners_count == 1 && (double) max_votes / votes_count > 0.5 ) )
    { return winners_count;
    }

return 0;
}

int find_losers(int candidates_count, int votes_count, int candidate_votes[20], int subroutine_results[20], bool losers[20])
{int min_votes = 1001;
int losers_count = 0;

for (int k=0; k<candidates_count; k++)
    {if (!losers[k]) {
        if (candidate_votes[k] < min_votes)
            {min_votes = candidate_votes[k];
            subroutine_results[0] = k;
            losers_count = 1;
        } else if (candidate_votes[k] == min_votes)
            { subroutine_results[(++losers_count-1)] =
              k;
            }
        }
    }

for (int i=0; i<losers_count; i++)
    { losers[subroutine_results[i]] = true;
    }

return losers_count;
}

void eliminate_loser(int remaining_candidates_count, int votes_count, int votes[][1000], int loser)
{for (int i = 0; i<remaining_candidates_count; i++) {
    for (int j = 0; j<votes_count; j++)
        {if (votes[i][j] == loser + 1) {
            for (int l = i; l<remaining_candidates_count; l++)
                {votes[l][j] = votes[l + 1][j];
                }
            }
        }
    }
}

int main( int argc, char * argv[] )
{int cases_count;
int candidates_count;
int remaining_candidates_count;
int votes_count;
char candidates[20][80];
int votes[20][1000];
int candidate_votes[20];
int subroutine_results[20];
bool losers[20];
int winners_count;
int losers_count;

```



```

std::cin >> cases_count;
std::cin.get();
std::cin.get();

for (int i=0; i<cases_count; i++) {
    parse_case(candidates_count, votes_count, candidates, votes);
    reset_losers(losers);
    remaining_candidates_count = candidates_count;
    winners_count = 0;
    while (!winners_count) {
        count_votes(candidates_count, votes_count, candidate_votes, votes);
        winners_count = find_winners(candidates_count, remaining_candidates_count, votes_count, candidate_votes,
subroutine_results);
        if (winners_count == 0) {
            losers_count = find_losers(candidates_count, votes_count, candidate_votes, subroutine_results, losers);
            for (int k = 0; k<losers_count; k++) {
                eliminate_loser(remaining_candidates_count, votes_count, votes, subroutine_results[k]);
                remaining_candidates_count--;
            }
        }
    }
    for (int j=0; j<winners_count; j++) {
        std::cout << candidates[subroutine_results[j]] << std::endl;
    }
    std::cout << std::endl;
}

return 0;
}

```

Output :

```

1
3
John doe
Jane Smith
Jane Austren
1 2 3
2 1 3
2 3 1
1 2 3
3 1 2

John doe

-----
Process exited after 110.1 seconds with return value 0
Press any key to continue . . .

```


Experiment 4


A) Jolly Jumper

Program:

```
#include <stdio.h>
int main(){
    static int n, i, j, v[100], a[100];

    while(scanf("%d",&n) ==
        1){for(i = 0; i < n; i++){
        scanf("%d",&v[i]);
        a[i] = 0;
        }
        j = n-1;
        for(i = 0; i < j; i++)
            a[(v[i]-v[i+1])] = 1;
        j = 1;
        for(i = 1; i < n;
            i++){if(!a[i]){
            j = 0;
            break;
            }
        }
        if(j)
            printf("Jolly\n");
        else
            printf("Not jolly\n");
    }
    return 0;
}
```

Output



```
4 1 4 2 3
Not jolly

5 1 4 2 -1 6
Not jolly
```

Experiment 4

B)Hartals

Program :

```
#include<stdio.h>
main()
{
int i,j,d,x,count_nwd,p,n,hp[100];
char arr[100][100];
printf("enter the number of days:\n"); scanf("%d",&d);
printf("\nenter the number of political parties:\n");
scanf("%d",&p);
printf("\nenter hartal parameters for %d parties\n",p);
for (x=0;x<p;x++)
{
scanf("%d",&hp[x]);
}
for(i=0;i<3 ;i++)
{
for(j=0;j<14;j++)
{
if((j+1)%hp[i]==0)
{
arr[i][j]='x';
}
else
{
arr[i][j]='0';
}
printf("%c",arr[i][j]);
}
printf("\n");
for(i=0;i<14;i++)
{
for(j=0;j<3;j++)
{
if(arr[j][i]=='x')
{
if(i%7==6 || i%7==0 )
{
break;
}
}
else
{
count_nwd++; break;
}
}
}
}
printf("\nnonworking days = %d\n",count_nwd);
```

}

Output :

```
enter the number of days:
14

enter the number of political parties:
3

enter hartal parameters for 3 parties
3
3
4
00x00x00x00x00
00x00x00x00x00
000x000x000x00

nonworking days = 6

-----
```

Experiment 4

C) Erdos number

Program :

```
#include <iostream>
#include <vector>
#include <string>
#include <sstream>
#include <fstream>
#include <stdio.h>    /* printf */
#include <stdlib.h>    /* abs */
#include <cmath>
#include <map>
#include <queue>
#include <functional>
#include <sstream>
#include <vector>
#include <algorithm>
using namespace std;
map<string,int> ErdosNumbers; map<string,bool> Visited;
void ComputeErdosNumbers(string FullName, map<string,vector<string> > CoAuthors, bool
start)
{
    vector<string> oldList, newList;
    for (int i=0; i<CoAuthors["Erdos, P."].size(); i++) ErdosNumbers[CoAuthors["Erdos, P."][i]] = 1;
    newList = CoAuthors["Erdos, P."];
    while(!newList.empty())
    {
        for (int i=0; i<newList.size(); i++)
        for (int j=0; j< CoAuthors[newList[i]].size(); j++)
        if (ErdosNumbers[CoAuthors[newList[i]][j]] > ErdosNumbers[newList[i]] + 1
        || ErdosNumbers[CoAuthors[newList[i]][j]] == -1)
        {
            ErdosNumbers[CoAuthors[newList[i]][j]] = ErdosNumbers[newList[i]] + 1;
            oldList.push_back(CoAuthors[newList[i]][j]);
        }
        newList = oldList; oldList.clear();
    }
}
vector < string > extract_name(string &line)
{
    vector < string > list; string::size_type begin(0);
    string::size_type end = line.find(".", begin); while (end != string::npos)
    {
        list.push_back(line.substr(begin, end - begin + 1)); begin = end + 3;
        end = line.find(".", begin);
    }
    if (begin < (line.length() - 1)) list.push_back(line.substr(begin));
    return list;
}
```

```

int main()
{
int N; scanf("%d\n",&N); for (int i=0; i<N; i++)
{
ErdosNumbers.clear(); int nAuthors, nPapers;
scanf("%d %d\n", &nPapers, &nAuthors); map<string,vector<string> > CoAuthors; char
PaperString[500000];
for (int j=0; j<nPapers; j++)
{
string Names; gets(PaperString); vector<string> PaperAuthors; stringstream s(PaperString);
getline(s,Names,':');
PaperAuthors = extract_name(Names);
for (int k = 0; k<PaperAuthors.size(); k++) ErdosNumbers[PaperAuthors[k]] = -1; for (int k=0;
k<PaperAuthors.size(); k++)
for (int h=0; h <PaperAuthors.size(); h++) if (PaperAuthors[h] != PaperAuthors[k])
{
if (find(CoAuthors[PaperAuthors[k]].begin(),
CoAuthors[PaperAuthors[k]].end(),PaperAuthors[h]) == CoAuthors[PaperAuthors[k]].end())
CoAuthors[PaperAuthors[k]].push_back(PaperAuthors[h]);
}
}
ErdosNumbers["Erdos, P."] = 0; char AName[500000]; vector<string> ANames;
for (int j=0; j<nAuthors; j++)
{
gets(AName); ANames.push_back(AName);
}
ComputeErdosNumbers("",CoAuthors,true); cout << "Scenario " << i+1 << endl;
for (int j=0; j<nAuthors; j++)
{
cout << ANames[j] << " ";
map<string,int>::iterator it = ErdosNumbers.find(extract_name(ANames[j])[0]); if (it ==
ErdosNumbers.end() || it->second == -1) cout << "infinity";
else cout << it->second; cout << endl;
}
}
return 0;
}

```

Output :

```

1
4 3
Smith, M.N., Martin, G., Erdos, P.: Newtonian forms of prime factors
Erdos, P., Reisig, W.: Stuttering in petri nets
Smith, M.N., Chen, X.: First order derivatives in structured programming
Jablonski, T., Hsueh, Z.: Selfstabilizing data structures
Smith, M.N.
Hsueh, Z.
Chen, X.
Scenario 1
Smith, M.N. 1
Hsueh, Z. infinity
Chen, X. 2
-----
Process exited after 104 seconds with return value 0
Press any key to continue . . .

```


Experiment 5

A) WERTYU

Program :

```
#include <stdio.h>

#include <string.h>

int findKey(char input, char key[60]);

int main(void)
{
    int i;
    int length;
    char input[500];

    char output[500];

    char keyboard[50] = { '^', '1', '2', '3', '4', '5', '6', '7', '8', '9', '0', '-', '=',
        'Q', 'W', 'E', 'R', 'T', 'Y', 'U', 'I', 'O', 'P', '[', ']', '\\',
        'A', 'S', 'D', 'F', 'G', 'H', 'J', 'K', 'L', ';',
        'Z', 'X', 'C', 'V', 'B', 'N', 'M', ',', '.', '/', };

    while( !feof(stdin) ){

        if( !gets(input) ) break;

        length = strlen(input);

        for(i=0; i<length; i++){

            if(input[i] == '^' || input[i] == 'Q' || input[i] == 'A' || input[i] == 'Z') output[i] = '*';

            else if( findKey(input[i], keyboard) == -1 ) output[i] = ' ';

            else output[i] = keyboard[findKey(input[i], keyboard)-1];

        }

        for(i=0; i<length; i++){

            if(output[i] == '*') continue;

            printf("%c", output[i]);

        }printf("\n");

        return 0;
    }

    int findKey(char input, char key[60])
    {
        int i;

        for(i=0; i<50; i++){
            if( input == key[i] ) return i;
        }
        return -1;}
}
```

Output :

```
O S, GOMR YPFSU/  
I AM FINE TODAY.
```

```
|
```

Experiment 5

B) Where is Waldrof ?

Program :

```
#include <cstdlib>
#include <iostream>
#include <string>
using namespace std;
bool check_word(string _words_[][2], int num_words, string find[], int *_pos_);
bool check_word(string _words_[][2], int num_words, string find[], int *_pos_)
{
    for(*_pos_ = 0; *_pos_ < num_words; *_pos_++)
    {
        for(int j = 0; j < 8; j++)
        {
            if(find[j].find(_words_[*_pos_][0]))
            {
                return true;
            }
        }
    }
    return false;
}
void find_word(int x, int y, int _n, int _m, string find[], char word_find[][11]);
void find_word(int x, int y, int _n, int _m, string find[], char word_find[][11])
{
    int temp_x = x;
    int temp_y = y;
    while(temp_x >= 1)//left
    {
        find[0] += word_find[temp_x][y];
        temp_x--;
    }
    temp_x = x;
    while(temp_x >= _n)//right
    {
        find[1] += word_find[temp_x][y];
        temp_x++;
    }
    temp_x = x;
    while(temp_y >= 1)//up
    {
        find[2] += word_find[x][temp_y];
        temp_y--;
    }
    temp_y = y;
    while(temp_y <= _m)//down
    {
        find[3] += word_find[x][temp_y];
        temp_y++;
    }
    temp_y = y;
    while((temp_x >= 1) && (temp_y >= 1))//up-left
    {
        find[4] += word_find[temp_x][temp_y];
        temp_x--;
        temp_y--;
    }
}
```

```

temp_x = x;
temp_y = y;
while((temp_x <= _n) &&(temp_y >= 1))//up-right
{
    find[5] += word_find[temp_x][temp_y];
    temp_x ++;
    temp_y --;
}
temp_x = x;
temp_y = y;
while((temp_x >= 1) &&(temp_y <= _m))//down--left
{
    find[6] += word_find[temp_x][temp_y];
    temp_x --;
    temp_y ++;
}
temp_x = x;
temp_y = y;
while((temp_x <= _n) &&(temp_y <= _m))//down-right
{
    find[7] += word_find[temp_x][temp_y];
    temp_x ++;
    temp_y ++;
}
temp_x = x;
temp_y = y;
cout << "inside find" << endl;
for(int i = 0; i < 8; i++)
{
    cout << find[i] << endl;
}
}
int main(int argc, char** argv)
{int n = 11;
int m = 8;
int x, y, pos = 0;
int num_words = 4;
string guess[8];
char word_find[8][11] =
{{'a','b','c','d','e','f','g','h','i','g','g'},{'h','e','b','k','w','a','l','d','o','r','k'},{'f','t','y','a','w','a','l','d','o','r','m'},{'f','t','s','i','m','r','l','q','s','r','c'},
{'b','y','o','a','r','b','e','d','e','y','v'},{'k','l','c','b','q','w','i','k','o','m','k'},{'s','t','r','e','b','g','a','d','h','r','b'},{'y','u','i','q','l','x','c','n','b','j','f'}}
;
string words[4][2] = {"waldorf","0"}, {"bambi","0"}, {"betty","0"}, {"dagbert","0"};
find_word(x, y, n, m, guess, word_find);
return 0;
}

```

Output :

Output Clear

```
/tmp/vkLqihN4DT.o
abcDEFGhigg
hEbkWalDork
FtyAwaldORm
FtsimrLqsrc
byoArBeDeyv
Klcbqwikomk
strEBGadhrb
yUiqlxcnBjf
2 5
2 3|
```

Experiment 5

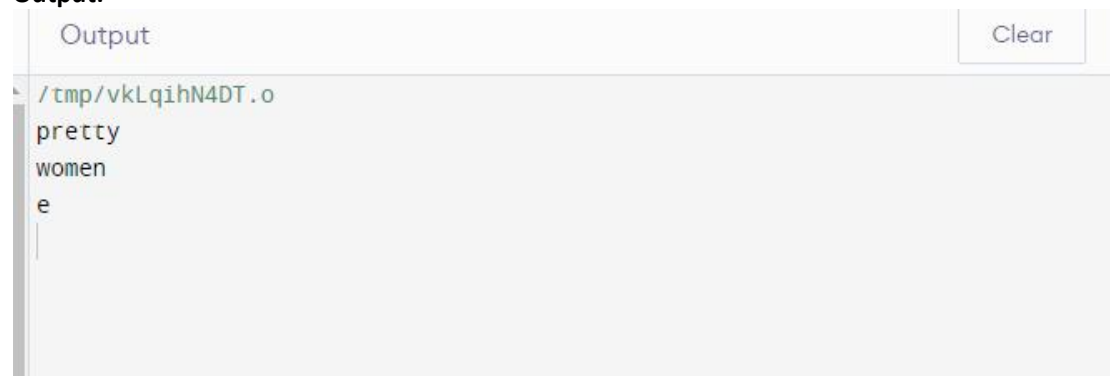
C) Common Permutaion

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

using namespace std;

int main()
{
    string a,b;
    while(getline(cin,a)){
        getline(cin,b);
        int aCount[26] = {};
        int bCount[26] = {};
        for(auto& c : a){
            aCount[c-'a']++;
        }
        for(auto& c : b){
            bCount[c-'a']++;
        }
        string res = "";
        for(int i=0;i<26;i++){
            int common = min(aCount[i],bCount[i]);
            for(int j=0;j<common;j++)
                res += (char)(i+'a');
        }
        cout << res << endl;
    }
}
```

Output:



The screenshot shows a code editor window with a tab labeled "Output". The output text is as follows:

```
/tmp/vkLqihN4DT.o
pretty
women
e
|
```

Experiment No: 6

Problems: A) Automated Judge Script

Program:

```
#include <stdio.h> #include <string.h> #include <iostream> using namespace std;
char A[105][150], B[105][150], bufA[10000], bufB[10000];
int n, m; int AC() {
    if(n != m) return 0; int i;
    for(i = 0; i < m; i++) if(strcmp(A[i], B[i]))
        return 0;
    return 1;
}
int PE() {
    int i, j, idxA = 0, idxB = 0; for(i = 0; i < n; i++) {
        for(j = 0; A[i][j]; j++) {
            if(A[i][j] >= '0' && A[i][j] <= '9')
                bufA[idxA++] = A[i][j];
        }
    }
    for(i = 0; i < m; i++) { for(j = 0; B[i][j]; j++) {
        if(B[i][j] >= '0' && B[i][j] <= '9')
            bufB[idxB++] = B[i][j];
        }
    }
    bufA[idxA] = '\0';
    bufB[idxB] = '\0';
    return !strcmp(bufA, bufB);
}
int judge() {
    if(AC()) return 1;
    if(PE()) return 2;
    return 3;
}
int main() {
    int cases = 0;
    while(scanf("%d", &n) == 1 && n) { cin.ignore(100, '\n');
        int i;
        for(i = 0; i < n; i++) gets(A[i]);
```

```
scanf("%d", &m);
cin.ignore(100, '\n'); for(i = 0; i < m; i++)
gets(B[i]);
int flag = judge(); printf("Run #%d: ", ++cases); if(flag == 1)
puts("Accepted"); else if(flag == 2)
puts("Presentation Error"); else
puts("Wrong Answer");
}
return 0;
}
```

Output:

```
2
10
5
2
10
15
Run #1: Wrong Answer
2
```


Experiment No: 6

Problems: B) File Fragmentation

Program:

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    int t;
    string in;
    scanf("%d\n\n",&t);
    while(t--){
        vector<string> parts;
        int minSize = INT_MAX, maxSize = INT_MIN;
        while(getline(cin,in), !in.empty()){
            minSize = min((int)in.length(),minSize);
            maxSize = max((int)in.length(),maxSize);
            parts.push_back(in);
        }
        unordered_map<string,int> cntMap;
        string best = "";
        for(int i=0;i<parts.size();i++)
            for(int j=i+1;j<parts.size();j++)
                if(parts[i].size() + parts[j].size() == (minSize+maxSize)){
                    string combine1 = parts[i]+parts[j], combine2 = parts[j]+parts[i];
                    if(++cntMap[combine1] > cntMap[best]) best = combine1;
                    if(++cntMap[combine2] > cntMap[best]) best = combine2;
                }
        cout << best << endl;
        if(t) cout << endl;
    }
}
```

Output:



```
1
011
0111
01110
111
0111
10111
01110111
```

Experiment No: 6

Problems: C) Doublets

Program:

```
#include <set>
#include <map>
#include <list>
#include <cmath>
#include <ctime>
#include <climits>
#include <queue>
#include <stack>
#include <cctype>
#include <cstdio>
#include <string>
#include <vector>
#include <cassert>
#include <cstdlib>
#include <cstring>
#include <sstream>
#include <iostream>
#include <algorithm>
using namespace std;
#define FOR(i, L, U) for(int i=(int)L; i<=(int)U; i++)
#define FORD(i, U, L) for(int i=(int)U; i>=(int)L; i--)
#define READ(x) freopen(x, "r", stdin)
#define WRITE(x) freopen(x, "w", stdout)
#define PQ priority_queue
#define PB push_back
#define SZ size()
#define ff first
#define ss second
#define EPS 1e-9
#define SQR(x) ((x)*(x))
#define INF 99999999
#define ALL_BITS ((1 << 31) - 1)
#define NEG_BITS(mask) (mask ^ ALL_BITS)
#define TEST_BIT(mask, i) (mask & (1 << i))
#define ON_BIT(mask, i) (mask |= (1 << i))
```

```

#define OFF_BIT(mask, i) (mask &= NEG_BITS(1 << i))
typedef long long LL;
typedef vector<char> VC;
typedef vector<vector<char> > VVC;
typedef vector<int> VI;
typedef vector<vector<int> > VVI;
typedef vector<string> VS;
typedef vector<bool> VB;
typedef vector< vector<bool> > VVB;
typedef pair<int, int> PII;
typedef map<int, int> MII;
typedef map<char, int> MCI;
typedef map<string, int> MSI;
typedef map<int, string> MIS;
#define WHITE 0
#define GRAY 1
#define BLACK 2
#define MAX_NODE 25145
string name;
int nodes;
int dist[MAX_NODE];
bool color[MAX_NODE];
int pre[MAX_NODE];
int u,v;
MIS rev;
MSI dic;
void bfs(int src){
    map<int,string> ::iterator revEnd = rev.end();
    string ustr,vstr;
    queue<int> q;
    FOR(i,1,nodes){
        dist[i] = INF;
        color[i] = false;
        pre[i] = i;
    }
    dist[src] = 0;
    color[src] = true;
    q.push(src);

```

```

while(!q.empty()){
    u = q.front();
    ustr = rev[u];
    q.pop();
    FOR(i,0,ustr.length()-1){
        FOR(j,'a','z'){
            if(ustr[i]==j)continue;
            vstr= ustr;
            vstr[i] = j;
            v = dic[vstr];
            if(color[v]==false&&rev.find(v)!=revEnd){
                color[v] = true;
                dist[v] = dist[u] + 1;
                pre[v] = u;
                q.push(v);
            }
        }
    }
}

int main()
{
    //READ("input.txt");
    //WRITE("output.txt");
    map<string,int> ::iterator it;
    string str;
    string st,en;
    bool letBlank = false;
    while(getline(cin,name)){
        if(name!=""){
            if(!dic[name]){
                dic[name] = ++nodes;
                rev[nodes] = name;
            }
        }
        else while(cin >> st >> en){
            if(letBlank) printf("\n");
            letBlank = true;

```

```

    bfs(dic[st]);
    if(dist[dic[en]]==INF)printf("No solution.\n");
    else {
        VI path;
        int u,v;
        v = dic[en];
        u = dic[st];
        while(v!=u){
            path.push_back(v);
            v = pre[v];
        }
        path.push_back(u);
        FORD(i,path.size()-1,0) cout << rev[path[i]] << endl;
    }
}
}
return 0;
}

```

Output:

Output

Clear

```

poasted
hoasted
foasted

roaster
coaster
coastal
posted
poasted
roaster

coaster

coastal

posted

poasted

No solution.
|

```

Activate Windows

Go to Settings to activate Windows.

Experiment No: 7

Problem: A) Vito's Family

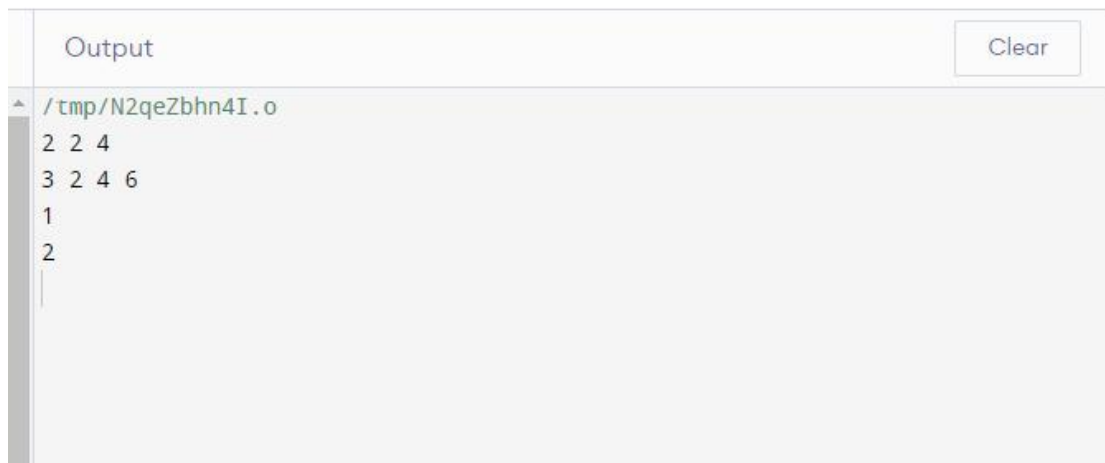
Program:

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    int t,n,v;
    cin >> t;
    while(t--){
        cin >> n;
        vector<int> neigh;
        int sum = 0;
        while(n--){
            cin >> v;
            neigh.push_back(v);
        }
        sort(neigh.begin(),neigh.end());
        int median = neigh[neigh.size()/2];

        for(auto& v : neigh) sum += abs(v-median);

        cout << sum << endl;
    }
}
```

Output:



The screenshot shows a window titled "Output" with a "Clear" button in the top right corner. The output text is as follows:

```
/tmp/N2qeZbhn4I.o
2 2 4
3 2 4 6
1
2
```

Experiment No: 7

B) Bridge

Program:

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <string>
#include <cctype>
#include <stack>
#include <queue>
#include <list>
#include <vector>
#include <map>
#include <sstream>
#include <cmath>
#include <bitset>
#include <utility>
#include <set>
#include <numeric>
#include <time.h>
#include <fstream>
#define INT_MAX 2147483647
#define INT_MIN -2147483648
#define pi acos(-1.0)
#define E 2.71828182845904523536
using namespace std;
int main()
{
    int N,n; cin >> N;
    for (int i=0; i<N; i++)
    {
        deque<int> LeftSide, RightSide;
        cin >> n; int T;
        for (int i=0; i<n; i++)
        {
            cin >> T; LeftSide.push_back(T);
        }
        sort(LeftSide.begin(), LeftSide.end()); int TotalTime = 0;
        stringstream fout; while(1)
        {
            int A = LeftSide[0];
            if (LeftSide.size() == 1) {fout << A; TotalTime += A; break;} int B = LeftSide[1];
            if (LeftSide.size() == 2)
            {
                fout << A << " " << B; TotalTime += B;break;
            }
            if (LeftSide.size() == 3)
            {
                fout << A << " " <<LeftSide[2] << endl << A << endl << A << " " << B ; TotalTime += B + A +
                LeftSide[2];
                break;
            }
            int Y, Z;
            Z = LeftSide.back(); LeftSide.pop_back(); Y = LeftSide.back(); LeftSide.pop_back(); if (A + Y <
            2*B)
```

```

{
fout << A << " " << Z << endl << A << endl << A << " " << Y << endl << A << endl;
TotalTime += Z + 2*A + Y;
}
else
{
fout << A << " " << B << endl << A << endl << Y << " " << Z << endl << B << endl; TotalTime
+= 2*B + A + Z;
}
}
cout<< TotalTime << endl << fout.str() << endl;
if (i != N-1) cout << endl ;
}
return 0;
}
Output:

```

```

1
4
1
2
5
10
17
1 2
1
5 10
2
1 2

```


Experiment no : 8

A] Longest Nap:

Code:

```
#include <stdio.h>
#include <stdlib.h>
typedef struct {
    int st, ed;
} Time;
int cmp(const void *i, const void *j) {
    Time *a, *b;
    a = (Time *)i, b = (Time *)j;
    return a->st - b->st;
}
int main() {
    int n, i, a, b, c, d, day = 0;
    Time D[100];
    while(scanf("%d", &n) == 1) {
        for(i = 0; i < n; i++) {
            scanf("%d:%d %d:%d", &a, &b, &c, &d);
            D[i].st = a*60 + b;
            D[i].ed = c*60 + d;
            while(getchar() != '\n');
        }
        qsort(D, n, sizeof(Time), cmp);
        int lastT = 600, ans = 0, st;
        for(i = 0; i < n; i++) {
            if(abs(lastT - D[i].st) > ans)
                ans = abs(lastT - D[i].st), st = lastT;
            lastT = D[i].ed;
        }
        if(abs(18*60 - lastT) > ans)
            ans = abs(18*60 - lastT), st = lastT;
        printf("Day #%d: the longest nap starts at ", ++day);
        printf("%02d:%02d and will last for ", st/60, st%60);
        if(ans >= 60)
            printf("%d hours and ", ans/60);
        printf("%d minutes.\n", ans%60);
    }
    return 0;
}
```

Output:

```

/tmp/KbVsscrMqR.o
4
10:00 12:00
12:00 13:00
13:00 15:00
15:30 17:45
Day #1: the longest nap starts at 15:00 and will last for 30 minutes.
|
```

B] Football

Code:

```
#include <iostream>
#define INT_MAX 2147483647
#define INT_MIN
-2147483648
#define pi acos(
-1.0)
#define E 2.71828182845904523536
using namespace std;
struct Team {
int nGames, nWins,nTies,nLoss, nGoals, nGoalsAgainst, nPoints;
string Name;
Team(string TN)
{
nGames = nWins = nTies = nLoss = nGoals = nGoalsAgainst = 0;
Name = TN;
}
Team()
{
nGames = nWins = nTies = nLoss = nGoals = nGoalsAgainst = 0;
}
};
bool CompareTeams(pair<string,Team> e1, pair<string,Team> e2)
{
Team a = e1.second;
Team b = e2.second;
int PointsA = a.nWins*3 + a.nTies;
int PointsB = b.nWins*3 + b.nTies;
int GDA = a.nGoals - a.nGoalsAgainst;
int GDB = b.nGoals - b.nGoalsAgainst;
if (PointsA != PointsB) return PointsA > PointsB;
if (a.nWins != b.nWins) return a.nWins > b.nWins;
if (GDA != GDB) return GDA > GDB;
if (a.nGoals != b.nGoals) return a.nGoals > b.nGoals;
if (a.nGames != b.nGames) return a.nGames < b.nGames;
for(int i=0; i<(int)a.Name.length(); i++) a.Name[i] =
tolower(a.Name[i]);
for(int i=0; i<(int)b.Name.length(); i++) b.Name[i] =
tolower(b.Name[i]);
return a.Name < b.Name;
}
int main()
{
```

```

int N;
scanf("%d\n",&N);
for (int i=0; i<N; i++)
{
    int nTeams;
    char TournName[1000];
    gets(TournName);
    scanf("%d\n",&nTeams);
    map<string, Team> Teams;
    for (int j=0; j<nTeams; j++)
    {
        char TeamName[1000];
        gets(TeamName);
        Team T(TeamName);
        Teams.insert(make_pair(TeamName,T));
    }
    int nGames;
    scanf("%d",&nGames);
    char Game[1000];
    gets(Game);
    for (int j=0; j<nGames; j++)
    {
        gets(Game);
        char buff;
        stringstream s(Game);
        string TeamNameA, TeamNameB;
        int GoalsA, GoalsB;
        getline(s,TeamNameA,'#');
        s >> GoalsA >> buff >> GoalsB >> buff;
        getline(s,TeamNameB);
        Teams[TeamNameA].nGames++;
        Teams[TeamNameB].nGames++;
        Teams[TeamNameA].nGoals+= GoalsA;
        Teams[TeamNameB].nGoals+= GoalsB;
        Teams[TeamNameA].nGoalsAgainst += GoalsB;
        Teams[TeamNameB].nGoalsAgainst += GoalsA;
        if (GoalsA == GoalsB) Teams[TeamNameA].nTies++,
        Teams[TeamNameB].nTies++;
        else if (GoalsA > GoalsB) Teams[TeamNameA].nWins++,
        Teams[TeamNameB].nLoss++;
        else Teams[TeamNameA].nLoss++, Teams[TeamNameB].nWins++;
    }
    vector< pair<string, Team> > SortedTeams;
    copy(Teams.begin(), Teams.end(), back_inserter(SortedTeams));
}

```

```

sort(SortedTeams.begin(), SortedTeams.end(), CompareTeams);
printf("%s\n", TournName);
for (int j=0; j<SortedTeams.size(); j++)
printf("%d) %s %dp, %dg (%d-%d-%d), %dgd (%d-%d)\n",
j+1, SortedTeams[j].first.c_str(), SortedTeams[j].second.nWins*3 +
SortedTeams[j].second.nTies,
SortedTeams[j].second.nGames,
SortedTeams[j].second.nWins, SortedTeams[j].second.nTies,
SortedTeams[j].second.nLoss,
SortedTeams[j].second.nGoals - SortedTeams[j].second.nGoalsAgainst,
SortedTeams[j].second.nGoals,
SortedTeams[j].second.nGoalsAgainst);
if (i != N-1) printf("\n");
}
return 0;
}

```

Output:

```

Brazil
Norway
Morocco
Scotland
5
Brazil#2@1#Scotland
Norway#2@2#Morocco
Scotland#1@1#Norway
Brazil#3@0#Morocco
Morocco#3@0#Scotland
Brazil#1@2#Norway
World Cup 1998 - Group A
1) Brazil 6p, 3g (2-0-1), 3gd (6-3)
2) Norway 5p, 3g (1-2-0), 1gd (5-4)
3) Morocco 4p, 3g (1-1-1), 0gd (5-5)
4) Scotland 1p, 3g (0-1-2), -4gd (2-6)

Some strange tournament
5
Team A
Team B
Team C
Team D
Team E
5
Team A#1@1#Team B
Team A#2@2#Team C
Team A#0@0#Team D
Team E#2@1#Team C
Team E#1@2#Team D

```

A] Primary Arithmetic:

Code:

```
#include <iostream>
#define INT_MAX 2147483647
#define INT_MIN -2147483648
#define pi acos(-1.0)
#define E 2.71828182845904523536
using namespace std;
int main()
{
while(1)
{
int m,n;
72
cin >> m >> n;
if (m == 0 && n == 0) break;
int carry=0, ncarries =0;
while( m > 0 || n > 0)
{
carry = (m%10 + n%10 + carry)/10;
m/= 10;
n /= 10;
if (carry) ncarries++;
}
if (ncarries == 0) cout << "No carry operation.\n";
else cout << ncarries << " carry operation" << ((ncarries > 1) ?
("s.\n") : (".\n"));
}
return 0;
}
```

Output:

```
123 456
No carry operation.
555 555
3 carry operations.
123 594
1 carry operation.
0 0
```

B] Reverse and add

Code:

```
#include <iostream>
#define INT_MAX 2147483647
#define INT_MIN
-2147483648
#define pi acos(
-1.0)
#define E 2.71828182845904523536
using namespace std;
long long int reverInt(long long int x) {
    long long int rn=0;
    while(x > 0) {
        rn*= 10;
        rn += x%10;
        x/= 10;
    }
    return rn;
}
int main() {
    int N;
    cin >> N;
    for (int i=0; i<N; i++) {
        long long int n, rn=0, nreversals=0;
        cin >> n;
        while(reverInt(n) != n) {
            nreversals++;
            n += reverInt(n);
        }
        cout << nreversals << " " << n << endl;
    }
    return 0;
}
```

Code:

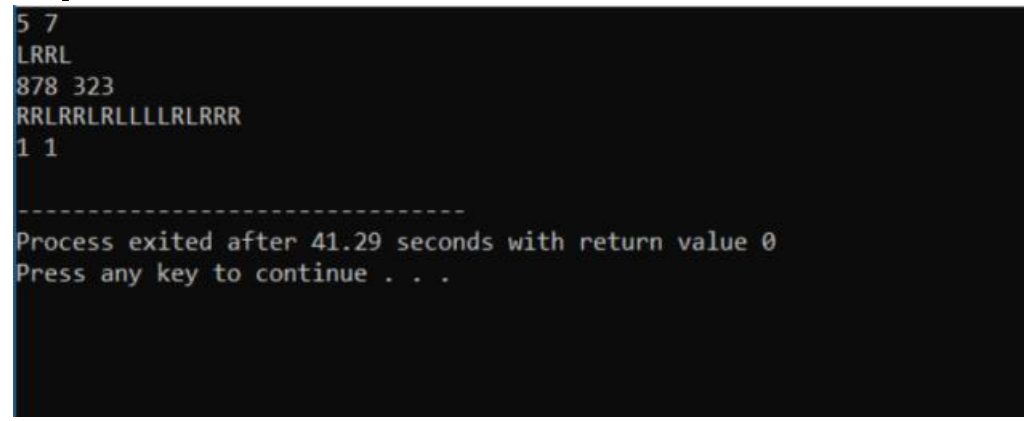
```
3
195
4 9339
265
5 45254
750
3 6666
-----
Process exited after 11.46 seconds with return value 0
Press any key to continue . . .
```

C] The stern-brocot number system

Code:

```
#include<stdio.h>
int A, B;
int a, b, c, d, e, f;
int main () {
    for(;;) {
        scanf("%d %d", &A, &B);
        if(A == 1 && B == 1) break;
        a = 0, b = 1, c = 1, d = 0;
        e = 1, f = 1;
        for(;;) {
            if(e == A && f == B) break;
            if(e * B > f * A) {
                printf("L");
                b = e, d = f;
            } else {
                printf("R");
                a = e, c = f;
            }
            e = a + b; f = c + d;
        }
        printf("\n");
    }
    return 0;
}
```

Output:



```
5 7
LRRRL
878 323
RRRLRRRLRRLLLRRLRRR
1 1

-----
Process exited after 41.29 seconds with return value 0
Press any key to continue . . .
```

A]How many Fibs ?

Code:

```
#include<stdio.h>
#include<string.h>
#define MAX(x,y) ( (x) >= (y) ? (x) : (y) )
char str1[10000],str2[10000],str3[10000];
char a[10000],b[10000];
bool bigger(char t1[],char t2[]){
    int len1=strlen(t1);
    int len2=strlen(t2);
    if(len1>len2)
        return true;
    else if(len1<len2)
        return false;
    else{
        for(int i=0;i<len1;i++)
            if(t1[i]>t2[i])
                return true;
            else if(t1[i]<t2[i])
                return false;
        return true;
    }
}
bool contain(char str[]){
    if(bigger(str,a)&&bigger(b,str))
        return true;
    else
        return false;
}
void plus(char str1[],char str2[],char str3[]){
    int len1=strlen(str1);
    int len2=strlen(str2);
    int len3=MAX(len1,len2)+1;
    int i,j,k,temp,carry;
    str3[len3]='\0';
    for(i=len1-1,j=len2-1,k=len3-1,carry=0;i>=0||j>=0;){
        if(i>=0&&j>=0)
            temp=str1[i--]-'0'+str2[j--]-'0'+carry;
        else if(i>=0)
            temp=str1[i--]-'0'+carry;
```



```

else
temp=str2[j--]-'0'+carry;
carry=temp/10;
temp%=10;
str3[k--]='0'+temp;
}
str3[k]='0'+carry;
if(str3[0]=='0')
memmove(str3, str3+1, sizeof(char)*len3);
}
int main() {
while(scanf("%s%s", a, b)==2) {
if(a[0]=='0' && b[0]=='0')
break;
str1[0]='1', str1[1]='\0';
str2[0]='2', str2[1]='\0';
int count=0;
if(contains(str1))
count++;
if(contains(str2))
count++;
while(!bigger(str2, b)) {
plus(str1, str2, str3);
if(contains(str3))
count++;
memmove(str1, str2, sizeof(str1));
memmove(str2, str3, sizeof(str2));
}
printf("%d\n", count);
}
return 0;
}

```

Output:

```

10 100
5
1234657890 9876543210
4
0 0

-----
Process exited after 40.16 seconds with return value 0
Press any key to continue . . .

```

B] How Many Pieces of Land?

Code:

```
#include <stdio.h>
```

```
int main() {  
    long long n;  
    scanf("%d");  
    while(scanf("%lld", &n) == 1)  
        printf("%lld\n", n*(n-1)*(n-2)*(n-3)/24 + n*(n-1)/2 + 1);  
    return 0;  
}
```

output:



The screenshot shows a terminal window with a blue title bar and a light gray background. The title bar text is "/tmp/KbVsscrMqR.o". The terminal displays the following output for inputs 4, 1, 1, 2, 2, 3, 4, 4, and 8:

```
4  
1  
1  
2  
2  
3  
4  
4  
8
```

C] Counting

Code:

```
#include<iostream>
using namespace std;
int countWays(int num)
{
    int dp[num+1];
    const int MOD = 1e9 + 7;
    dp[1] = 2;
    for(int i = 2; i <= num; ++i)
    {
        dp[i] = 0;
        for(int j = 1; j <= 3; ++j)
        {
            if(i - j == 0)
                dp[i] += 1;
            else if (j == 1)
                dp[i] += dp[i-j] * 2;
            else if(i - j > 0)
                dp[i] += dp[i-j];
            if(dp[i] >= MOD)
                dp[i] %= MOD;
        }
    }
    return dp[num];
}

int main()
{
    int n;
    cin>>n;
    cout << countWays(n);
    return 0;
}
```

Output:



```
/tmp/wKuFM22o22.o
3
13|
```