Week 06

=====================================================

Problem 601:

=====================================================

#include <iostream>

using namespace std;

// create function which find minimum number of a and b

int min(int a, int b) {

if (a < b)

return a;

else

return b;

}

int main() {

// create variables and input them

int a, b, c, d;

cin >> a >> b >> c >> d;

//output minimum of 4 numbers

cout << min(min(a, b), min(c, d));

return 0;

}

=====================================================

Problem 602:

=====================================================

#include <iostream>

using namespace std;

// create function which find power of number

double power(double a, int n)

{

double s = 1;

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

s \*= a;

return s;

}

int main()

{

// create variables and input them

int n;

double a;

cin >> a >> n;

//output power of number

cout << power(a, n);

return 0;

}

=====================================================

Problem 603:

=====================================================

#include <iostream>

using namespace std;

// function that realizes the function of "exclusive OR"

bool myXor(bool x, bool y)

{

if ((x == 1 && y == 0) || (x == 0 && y == 1))

return 1;

else

return 0;

}

int main()

{

// create variables and input them

int a, b;

cin >> a >> b;

// output 1 or 0

cout << myXor(a, b);

return 0;

}

=====================================================

Problem 604:

=====================================================

#include <iostream>

using namespace std;

// create function which return 1 or 0 which is most common among the values of its arguments x, y, z.

bool election(bool x, bool y, bool z)

{

int total = 0;

int massiv[] = { x,y,z };

for (int i = 0; i < 3; i++)

(massiv[i] == 1) ? total++ : total;

if (total >= 2)

return 1;

else

return 0;

}

int main()

{

// create variables and input them

int a, b, c;

cin >> a >> b >> c;

// output answer

cout << election(a, b, c);

return 0;

}

=====================================================

Problem 605:

=====================================================

#include <iostream>

using namespace std;

// creaate function which two given integers and return only "<" , ">" or "=".

char getSign(int a, int b)

{

if (a > b)

return '>';

else if (a < b)

return '<';

else

return '=';

}

int main()

{

// create variables and input them

int a, b;

cin >> a >> b;

// output answer

cout << getSign(a, b);

return 0;

}

=====================================================

Problem 606:

=====================================================

// create function which find out the array in which total number of positive elements is bigger.

void bigger()

{

int count\_1 = 0;

int count\_2 = 0;

cin >> N;

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

{

cin >> massiv[i];

(massiv[i] > 0) ? count\_1++ : count\_1;

}

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

{

cin >> massiv[j];

(massiv[j] > 0) ? count\_2++ : count\_1;

}

if (count\_1 > count\_2)

cout << "Number of positives in the first array is greater";

else if (count\_1 < count\_2)

cout << "Number of positives in the second array is greater";

else

cout << "Numbers are equal";

}

int main()

{

// function call

bigger();

return 0;

}

=====================================================

Problem 607:

=====================================================

#include <iostream>

using namespace std;

// create global varibales and array

int N, M;

int massiv[100][100];

int line = 0;

// create function which find out the row in which total number of positive elements is bigger.

void bigger()

{

int count\_1 = 0;

int count\_2 = 0;

bool x = false;

cin >> N >> M;

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

{

count\_1 = 0;

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

{

cin >> massiv[i][j];

if (massiv[i][j] > 0)

count\_1++;

}

if (count\_2 < count\_1)

{

count\_2 = count\_1;

line = i + 1;

x = false;

}

else if (count\_1 == count\_2)

x = true;

else

x = false;

}

(x == true) ? cout << "Numbers are equal" : cout << line;

}

int main()

{

// call function

bigger();

return 0;

}

=====================================================

Problem 608:

=====================================================

#include <iostream>

using namespace std;

// create global two-dimensional array

int N, M;

int massiv[100][100];

int line = 0;

// create function which find that finds the row in the given NxM matrix with longest sequence that contains same values.

void bigger()

{

int count\_1 = 0;

int count\_2 = 0;

bool x = false;

cin >> N >> M;

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

{

count\_1 = 0;

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

{

cin >> massiv[i][j];

if ((massiv[i][j] == massiv[i][j - 1]) && j != 0)

count\_1++;

}

if (count\_2 < count\_1)

{

count\_2 = count\_1;

line = i;

x = false;

}

else if (count\_1 == count\_2)

x = true;

else

x = false;

}

(x == true) ? cout << "No series of equal elements" : cout << "Longest series is in the string " << line;

}

int main()

{

// call function

bigger();

return 0;

}

=====================================================

Problem 609:

=====================================================

#include <iostream>

#include <cstring>

using namespace std;

const int SIZE = 200;

// create function which culculate how many times give letter meets int the given sequence

int count\_elements\_in\_array(char masssiv[SIZE], char element);

int main()

{

// create variables and arrays and input them

char a, b;

char massiv\_a[SIZE], massiv\_b[SIZE];

cin >> a >> b;

cin >> massiv\_a >> massiv\_b;

// output result

cout << count\_elements\_in\_array(massiv\_a, a) << " " << a << " characters in " << massiv\_a << endl;

cout << count\_elements\_in\_array(massiv\_b, b) << " " << b << " characters in " << massiv\_b << endl;

return 0;

}

int count\_elements\_in\_array(char massiv[SIZE], char element)

{

int result = 0;

for (int i = 0; i < strlen(massiv); i++)

if (massiv[i] == element)

result++;

return result;

}

=====================================================

Problem 610:

=====================================================

#include <iostream>

#include <cstring>

#include <string>

using namespace std;

const int SIZE = 200;

// create function which culculate how many times give letter meets int the given sequence

int count\_elements\_in\_array(string massiv, char element);

int main()

{

// create variables and arrays and input them

int N;

string massiv[SIZE];

char massiv\_2[SIZE];

cin >> N;

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

{

cin >> massiv\_2[i];

cin >> massiv[i];

}

// output answer

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

cout << count\_elements\_in\_array(massiv[i], massiv\_2[i]) << " " << massiv\_2[i] << " in " << massiv[i] << endl;

return 0;

}

int count\_elements\_in\_array(string massiv, char element)

{

int result = 0;

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

if (massiv[i] == element)

result++;

return result;

}

=====================================================

Problem 611:

=====================================================

#include <iostream>

#include <cstring>

#include <string>

using namespace std;

const int SIZE = 200;

//create function which to culculate how many times give letter meets int the given sequesnce.

int count\_elements\_in\_array(string massiv, char element);

int main()

{

// create variables and arrays and input them

int N;

string massiv[SIZE];

char massiv\_2[SIZE];

cin >> N;

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

{

cin >> massiv\_2[i];

cin >> massiv[i];

}

// output answer

for (int i = N - 1; i >= 0; i--)

cout << count\_elements\_in\_array(massiv[i], massiv\_2[i]) << " " << massiv\_2[i] << " in " << massiv[i] << endl;

return 0;

}

int count\_elements\_in\_array(string massiv, char element)

{

int result = 0;

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

if (massiv[i] == element)

result++;

return result;

}

=====================================================

Problem 612:

=====================================================

#include <iostream>

#include <cstring>

using namespace std;

// create global array

const int SIZE = 100;

char massiv[SIZE];

//create function that converts given sequence of characters from the lower case to upper case.

void up()

{

cin >> massiv;

for (int i = 0; i < strlen(massiv); i++)

(97 <= massiv[i] && massiv[i] <= 122) ? massiv[i] = massiv[i] - 32 : massiv[i];

cout << massiv;

}

int main()

{

//call function

up();

return 0;

}

=====================================================

Problem 613:

=====================================================

#include <iostream>

#include <cstring>

using namespace std;

//create global array

const int SIZE = 100;

char massiv[SIZE];

//create function which converts given sequence by the following rules:

//1 - If the letter is in upper case make it lower case;

//2 - If the letter is in lower case make it upper case;

//3 - If it's not a letter left it as is

void up()

{

cin >> massiv;

for (int i = 0; i < strlen(massiv); i++)

{

if (97 <= massiv[i] && massiv[i] <= 122)

massiv[i] = massiv[i] - 32;

else if (65 <= massiv[i] && massiv[i] <= 90)

massiv[i] = massiv[i] + 32;

}

cout << massiv;

}

int main()

{

//call function

up();

return 0;

}

=====================================================

Problem 614:

=====================================================

#include <iostream>

#include <cstring>

using namespace std;

// create global arrays

const int SIZE = 100;

char massiv[SIZE];

char massiv\_2[SIZE];

// create function which that converts given sequence by the following rules:

//1 - If the letter is in upper case make it lower case;

//2 - If the letter is in lower case make it upper case;

//3 - If it's not a letter left remove it from the sequence

void up()

{

int index = 0;

cin >> massiv;

for (int i = 0; i < strlen(massiv); i++)

{

if (97 <= massiv[i] && massiv[i] <= 122)

{

massiv\_2[index] = massiv[i] - 32;

index++;

}

else if (65 <= massiv[i] && massiv[i] <= 90)

{

massiv\_2[index] = massiv[i] + 32;

index++;

}

else

continue;

}

cout << massiv\_2;

}

int main()

{

// call function up

up();

return 0;

} =====================================================

Problem 615:

=====================================================

#include <iostream>

#include <cstring>

using namespace std;

// create function which return converting time in seconds

int time(int hours, int minute, int seconds)

{

int summ = 0;

summ = hours \* 3600 + minute \* 60 + seconds;

return summ;

}

// create function function which calculate the amount of time between two given times, both of which are within one 12-hour cycle of the clock.

void convertion(int time)

{

int hours, minute, seconds;

hours = time / 3600;

minute = (time % 3600) / 60;

seconds = (time % 3600) % 60;

(hours < 10) ? cout << 0 << hours : cout << hours;

cout << ":";

(minute < 10) ? cout << 0 << minute : cout << minute;

cout << ":";

(seconds < 10) ? cout << 0 << seconds : cout << seconds;

}

int main()

{

// create variables and input them

int time\_1;

int time\_2;

int h1, m1, s1, h2, m2, s2;

cin >> h1 >> m1 >> s1 >> h2 >> m2 >> s2;

// call functions and output answers

time\_1 = time(h1, m1, s1);

time\_2 = time(h2, m2, s2);

convertion(time\_2 - time\_1);

return 0;

}

=====================================================

Problem 616:

=====================================================

#include <iostream>

#include <string>

#include <math.h>

using namespace std;

//create global variable and array

string s;

int massiv[31];

//create function which converts the binary number to the decimal number.

int binary(int num)

{

int total = 0;

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

total += pow(2, i) \* massiv[num - i - 1];

return total;

}

int main()

{

// create variables and input them

int N;

cin >> N;

cin >> s;

int index = 0;

//convert string array to integer array

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

{

massiv[index] = s[i] - '0';

index++;

}

// output answer

cout << binary(N);

return 0;

}

=====================================================

Problem 617:

=====================================================

#include <iostream>

#include <cmath>

using namespace std;

//create global array

int massiv[100];

// create function that finds the minimum difference between adjacent elements of the array.

void SOLDAT(int n)

{

int j = 0;

int raznica = 10000;

int index\_i = 0;

int index\_j = 0;

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

{

j++;

if (raznica > abs((massiv[i] - massiv[j])))

{

raznica = abs(massiv[i] - massiv[j]);

index\_i = i;

index\_j = j;

}

if (j == n - 1)

j = -1;

}

cout << index\_i + 1 << " " << index\_j + 1;

}

int main()

{

// create variable and array and input them

int N;

cin >> N;

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

cin >> massiv[i];

//output answer

SOLDAT(N);

}

=====================================================

Problem 618:

=====================================================

#include <iostream>

#include <string>

#include <cstring>

using namespace std;

//create global array

int N;

char massiv[100];

//create function which checks the neighboring elements of the element

int checked(int n)

{

int j = 0;

bool x = true;

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

{

j++;

if ((massiv[i] == '+' || massiv[i] == '-' || massiv[i] == '/' || massiv[i] == '\*') && (massiv[j] == '+' || massiv[j] == '-' || massiv[j] == '/' || massiv[j] == '\*'))

{

x = false;

break;

}

if (j == n - 1)

j = -1;

}

return x;

}

int main()

{

// create variables and input them

int N;

cin >> N;

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

cin >> massiv[i];

// conditionality test

int x = 0;

int j = 0;

for (int i = 0; i < strlen(massiv); i++)

{

j++;

if (((48 <= int(massiv[i]) && int(massiv[i]) <= 57) || massiv[i] == '+' || massiv[i] == '-' || massiv[i] == '/' || massiv[i] == '\*') && checked(N) == 1)

x++;

else

break;

}

// output answer

(x == strlen(massiv)) ? cout << "YES" : cout << "NO";

return 0;

}