Week 08

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

Problem 801:

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

#include <iostream>

using namespace std;

void swap2(long\* number\_1, long\* number\_2)

{

long time\_variable;

time\_variable = \*number\_1;

\*number\_1 = \*number\_2;

\*number\_2 = time\_variable;

cout << \*number\_1 << " " << \*number\_2;

}

int main()

{

long number\_1, number\_2;

cin >> number\_1 >> number\_2;

swap2(&number\_1, &number\_2);

}

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

Problem 802:

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

#include <iostream>

using namespace std;

void Arnas(int\* mass\_Arnas, int N)

{

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

cin >> mass\_Arnas[i];

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

cout << mass\_Arnas[i] << " ";

}

int main()

{

int N;

cin >> N;

int\* mass\_Arnas = new int[N];

Arnas(mass\_Arnas, N);

return 0;

}

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

Problem 803:

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

#include <iostream>

#include <string>

#include <cstring>

using namespace std;

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

{

int result = 0;

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

if (massiv[i] == elements)

result++;

return result;

}

int main()

{

int N;

cin >> N;

char\* massiv\_of\_symbols = new char[N];

string\* massiv\_of\_strings = new string[N];

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

{

cin >> massiv\_of\_symbols[i];

cin >> massiv\_of\_strings[i];

}

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

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

Problem 804:

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

#include <iostream>

#include <cctype>

#include <cstring>

using namespace std;

void copy\_massiv(char\* Arnas)

{

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

if (isalpha(Arnas[i]) != 0)

cout << Arnas[i];

}

int main()

{

char Arnas[100];

cin >> Arnas;

copy\_massiv(Arnas);

}

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

Problem 806:

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

#include <iostream>

using namespace std;

int\* output(int\* massiv, int N)

{

int index = 0;

int\* answer = new int[N] {};

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

{

if ((i + 1) % 2 == 1)

{

answer[index] = massiv[i];

index++;

}

}

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

{

if ((i + 1) % 2 == 0)

{

answer[index] = massiv[i];

index++;

}

}

return answer;

}

int main()

{

int N;

cin >> N;

int\* Arnas = new int[N];

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

cin >> Arnas[i];

int\* answer\_massiv = output(Arnas, N);

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

cout << answer\_massiv[i] << " ";

}

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

Problem 807:

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

#include <iostream>

using namespace std;

int index = -1;

int index\_check(int\* massiv, int N)

{

index++;

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

if (massiv[i] == 0)

{

index = i;

break;

}

return index;

}

void output(int\* massiv, int N)

{

int total = 0;

int i = index\_check(massiv, N);

N = index\_check(massiv, N);

for (i; i < N; i++)

total += massiv[i];

cout << total;

}

int main()

{

int N;

cin >> N;

int\* massiv = new int[N];

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

cin >> massiv[i];

output(massiv, N);

}

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

Problem 808:

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

#include <iostream>

using namespace std;

int index\_check(int\* massiv, int N)

{

int index = 0;

int max = massiv[0];

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

if (massiv[i] >= max)

{

max = massiv[i];

index = i;

}

return index;

}

void final\_answer(int\* massiv, int N)

{

int total = 0;

N = index\_check(massiv, N);

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

(massiv[i] > 0) ? total += massiv[i] : total;

cout << total;

}

int main()

{

int N;

cin >> N;

int\* massiv = new int[N];

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

cin >> massiv[i];

final\_answer(massiv, N);

}

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

Problem 809:

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

#include <iostream>

#include <math.h>

using namespace std;

// find index maximum number in array

int index\_check\_max(int\* massiv, int N)

{

int index = 0;

int variable = massiv[0];

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

if (massiv[i] > variable)

{

variable = massiv[i];

index = i;

}

return index;

}

// find index minimum number in array

int index\_check\_min(int\* massiv, int N)

{

int index = 0;

int variable = massiv[0];

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

if (massiv[i] < variable)

{

variable = massiv[i];

index = i;

}

return index;

}

// find even number between min and max numbers

int output(int\* massiv, int N)

{

int total = 0;

for (int i = min(index\_check\_max(massiv, N), index\_check\_min(massiv, N)) + 1; i < max(index\_check\_max(massiv, N), index\_check\_min(massiv, N)); i++)

{

if (massiv[i] % 2 == 0)

total++;

}

return total;

}

int main()

{

int N;

cin >> N;

int\* massiv = new int[N];

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

cin >> massiv[i];

cout << output(massiv, N) << endl;

}

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

Problem 810:

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

#include <iostream>

#include <cstring>

using namespace std;

bool compare(char\* s1, char\* s2)

{

bool answer = true;

if (strlen(s1) == strlen(s2))

for (int i = 0; i < strlen(s1); i++)

{

if (s1[i] == s2[i])

answer = true;

else

{

answer = false;

break;

}

}

else

answer = false;

return answer;

}

int main()

{

char s1[100], s2[100];

cin >> s1 >> s2;

(compare(s1, s2) == true) ? cout << "YES" : cout << "NO";

}

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

Problem 811:

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

#include <iostream>

#include <cstring>

using namespace std;

bool compare(char\* string)

{

bool answer = true;

int j = strlen(string) - 1;

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

{

if (string[i] != string[j])

answer = false;

j--;

}

return answer;

}

int main()

{

char s[100];

cin >> s;

(compare(s)) ? cout << "YES" : cout << "NO";

}

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

Problem 812:

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

#include <iostream>

#include <cstring>

using namespace std;

char\* convert(char\* string, int step)

{

int length = strlen(string);

char\* answer = new char[length];

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

{

int variable = int(string[i]) - step;

if (variable < 65)

variable = 91 - (65 - variable);

answer[i] = variable;

}

return answer;

}

int main()

{

char string[100];

int step;

cin >> string >> step;

char\* answer = convert(string, step);

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

cout << answer[i];

return 0;

}

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

Problem 813:

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

#include <iostream>

#include <cstring>

using namespace std;

int check\_IP(int\* massiv, char\* symbols)

{

bool check\_number = true;

bool check\_symbols = true;

for (int i = 0; i < 4; i++)

{

if (0 <= massiv[i] && massiv[i] <= 255)

check\_number = true;

else

{

check\_number = false;

break;

}

}

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

{

if (symbols[i] == '.')

check\_symbols = true;

else

{

check\_symbols = false;

break;

}

}

if (check\_symbols == true && check\_number == true)

return 1;

else

return 0;

}

int main()

{

int massiv\_numbers[4];

char massiv\_symbols[4];

cin >> massiv\_numbers[0] >> massiv\_symbols[0] >> massiv\_numbers[1] >> massiv\_symbols[1] >> massiv\_numbers[2] >> massiv\_symbols[2] >> massiv\_numbers[3];

cout << check\_IP(massiv\_numbers, massiv\_symbols);

}

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

Problem 814:

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

#include <iostream>

using namespace std;

int main()

{

int steps;

string direction;

int x = 0, y = 0;

while (cin >> direction >> steps)

{

if (direction == "North")

y += steps;

if (direction == "South")

y -= steps;

if (direction == "West")

x -= steps;

if (direction == "East")

x += steps;

}

cout << x << " " << y;

return 0;

}

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

Problem 815:

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

#include <iostream>

#include <cstring>

using namespace std;

int check(int start, int last, char\* massiv)

{

int total = 0;

int Arnas = 0, luck\_ticket;

for (int i = start; i <= last; i++)

{

int l = i % 10000;

if (l == 0 || Arnas == 0)

{

Arnas = i / 10000;

luck\_ticket = massiv[Arnas / 100] + massiv[Arnas % 100];

}

total += (massiv[l / 100] + massiv[l % 100] == luck\_ticket);

}

return total;

}

int main()

{

char Arnas[100];

for (int i = 0; i <= 99; i++)

Arnas[i] = i / 10 + i % 10;

int first\_ticket, last\_ticket;

cin >> first\_ticket >> last\_ticket;

cout << check(first\_ticket, last\_ticket, Arnas);

}