SCENARIUSZ METODYCZNY ZAJĘĆ

PLANOWANYCH W KLASIE 2B

Temat: Funkcje w C++

Co uczestnik powinien już wiedzieć:

Uczestnik zna pojęcia:

* algorytm, pętle, zmienna, funkcja

Cele zajęć:

Uczestnik powinien:

* napisać proste programy w C++ z wykorzystaniem funkcji

Formy organizacyjne:

* praca równym frontem, praca samodzielna, praca w grupach

Środki dydaktyczne:

* komputer, rzutnik

Plan/ Przebieg zajęć:

1. **Część wstępna**
   1. Czynności organizacyjno- porządkowe.
2. **Część zasadnicza**
   1. Przypomnienie dotyczące funkcji void.
   2. Rozwiązanie zadania konkursowego.

**Tok lekcji:**

N: Na dzisiejszych zajęciach przypomnimy sobie co to jest funkcja void, a potem spróbujemy rozwiązać dość trudne zadanie konkursowe. Co to jest void?

U: Funkcja typu void nie zwraca żadnych danych, które można byłoby poddać dalszej obróbce. Funkcja tego typu może wykonywać pewne czynności, ale nie przekazuje informacji zwrotnej. Słowo kluczowe void oznacza "pusty", informuje, że nic nie będzie zwracane.

N: A teraz wykonajmy następujące zadanie:

Bajtazar dawno temu kupił spory obszar pola. Pole to było podzielone na kwadraciki jednostkowe, wśród których każde było albo użytkowe, albo nieużytkowe. Bajtazar pragnął rozpocząć uprawę na prostokątnej działce będącej częścią tego pola i złożonej tylko z kwadracików użytkowych∗ Od tego czasu minęło już prawie piętnaście lat. Bajtazar niedawno sprzedał stary obszar pola i kupił nowy, o wiele większy. Chciałby teraz na zakupionym terenie stworzyć dowolną prostokątną działkę złożoną jedynie z kwadracików użytkowych. Na ile sposobów może to zrobić?

Komunikacja z biblioteką

Aby móc rozwiązywać zadanie, należy na początku programu dopisać dyrektywę

#include "dzialka.h"

Dodana biblioteka udostępnia następujące funkcje:

• int GetFieldHeight(); – zwraca wysokość zakupionego pola.

• int GetFieldWidth(); – zwraca szerokość zakupionego pola.

• int IsUsableCell(int r, int c); – dla podanych 0 ≤ r < GetFieldHeight(), 0 ≤ c < GetFieldWidth() zwraca 1, jeśli kwadracik o współrzędnych (r, c) jest użytkowy lub 0 w przeciwnym przypadku.

N:A oto pełne rozwiązanie:

#include <bits/stdc++.h>

using namespace std;

template<class Function>

void for\_each\_div(int n, Function&& f) {

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

if (n % i == 0) {

f(i);

if (i \* i != n) {

f(n / i);

}

}

}

}

int solve1(int n) {

n--;

int res = 0;

auto f = [&res, n](int d) {

int r = n - d;

res += r > d && r % d == 0;

};

for\_each\_div(n, std::move(f));

return res;

}

int solve(int n) {

int res = solve1(n);

for\_each\_div(n, [&res](int d) { res += solve1(d); });

return res;

}

int main() {

int n;

scanf("%d", &n);

printf("%d\n", solve(n));

return 0;

}

//uczestnicy próbują samodzielnie pisać rozwiązanie, nauczyciel na bieżąco daje podpowiedzi.

Sfinansowane ze środków UE. Wyrażone poglądy i opinie są jedynie opiniami autora lub autorów i niekoniecznie odzwierciedlają poglądy i opinie Unii Europejskiej lub Europejskiej Agencji Wykonawczej ds. Edukacji i Kultury (EACEA). Unia Europejska ani EACEA nie ponoszą za nie odpowiedzialności.