Structură decizională, noiembrie 2024
Cls 8

NUMERE PARE ŞI IMPARE
Condiţie – structură decizională/alternativă

Varianta simplă:
Dându-se un număr natural n, să se determine dacă este par sau impar, fără a folosi operatorul % (modulo).
Exemplu. Pentru n = 11, răspunsul este IMPAR.
Explicarea metodei[1]

Această metodă de rezolvare se foloseşte de operatorii pe biţi. Ultimul bit al unui număr este 1, dacă n este impar, respectiv 0 în caz contrar (acest lucru se demonstrează uşor: toate celelalte puteri de 2 din reprezentarea binară sunt numere pare, mai puţin cel de pe poziţia 0, care este 1 — astfel, dacă ultimul bit al numărului este 1, atunci numărul este impar, respectiv par în caz contrar).

Aşadar, trebuie doar să aflăm valoarea ultimului bit al lui n. Acest lucru se poate face cu **operatorul AND(&)**, între numărul n şi 1. Aplicând această operaţie, vom obţine un număr care elimină toţi ceilalţi biţi din n, în afară de primul bit. Astfel, dacă numărul este 1, atunci n este impar, altfel este par.

```
#include <iostream>
using namespace std;
int main()
{
   //Declarăm şi citim numărul n
   int n;
   cin >> n;
   //Verificăm paritatea lui n
   if(n & 1) {
      cout << "IMPAR";
   } else {
      cout << "PAR";
   }
   return 0;
}
```

Aveţi de verificat programul cu unul dintre cele două compilatoare, apoi să rezolvaţi problema în Scratch.

Varianta cu modulo

```
#include <iostream>

using namespace std;

int main()
{
   int n;
   cout << "Scrie numărul dorit:"<<endl;
   cin >> n;
   //Verificam paritatea lui n
   if(n % 2) {
      cout << "IMPAR";
   } else {
      cout << "PAR";
   }
   return 0;
}
```

Aveţi de verificat programul cu unul dintre cele două compilatoare, apoi modificaţi programul în Scratch cu n mod 2.

[1] https://infoas.ro/lectie/82/verifica-daca-un-numar-este-par-sau-impar-fara-modulo-in-c