

Foaie cu Probleme Moderate - Subprograme in C++ (cu main)

1. Suma cifrelor pare

Scrie un subprogram care primeste un numar natural si returneaza suma cifrelor pare ale acestuia.

Exemplu input: n = 2486

Exemplu output: Suma cifrelor pare: 20

Main (exemplu):

```
#include <iostream>
using namespace std;

// Subprogramul va fi implementat aici

int main() {
    // Exemplu de apel
    n = 2486
    cout << "Suma cifrelor pare: " << /* apel subprogram */ << endl;
    return 0;
}
```

2. Numar prim

Scrie un subprogram care verifica daca un numar dat este prim.

Exemplu input: n = 17

Exemplu output: Numarul este prim: DA

Main (exemplu):

```
#include <iostream>
using namespace std;

// Subprogramul va fi implementat aici

int main() {
    // Exemplu de apel
    n = 17
    cout << "Numarul este prim: " << /* apel subprogram */ << endl;
    return 0;
}
```

3. Cel mai mare divizor comun

Scrie un subprogram care primeste doua numere si returneaza cel mai mare divizor comun al acestora.

Exemplu input: a = 24, b = 36

Exemplu output: Cel mai mare divizor comun: 12

Main (exemplu):

```
#include <iostream>
using namespace std;

// Subprogramul va fi implementat aici

int main() {
    // Exemplu de apel
    a = 24, b = 36
    cout << "Cel mai mare divizor comun: " << /* apel subprogram */ << endl;
    return 0;
}
```

4. Factorial

Scrie un subprogram care primește un număr natural și returnează factorialul acestuia.

Exemplu input: n = 5

Exemplu output: Factorial: 120

Main (exemplu):

```
#include <iostream>
using namespace std;

// Subprogramul va fi implementat aici

int main() {
    // Exemplu de apel
    n = 5
    cout << "Factorial: " << /* apel subprogram */ << endl;
    return 0;
}
```

5. Cifre în ordine crescătoare

Scrie un subprogram care verifică dacă cifrele unui număr sunt în ordine crescătoare.

Exemplu input: n = 1235

Exemplu output: Cifre în ordine crescătoare: DA

Main (exemplu):

```
#include <iostream>
using namespace std;

// Subprogramul va fi implementat aici

int main() {
    // Exemplu de apel
    n = 1235
    cout << "Cifre în ordine crescătoare: " << /* apel subprogram */ << endl;
    return 0;
}
```

6. Număr palindrom

Scrie un subprogram care verifică dacă un număr este palindrom.

Exemplu input: n = 1221

Exemplu output: Numărul este palindrom: DA

Main (exemplu):

```
#include <iostream>
using namespace std;

// Subprogramul va fi implementat aici

int main() {
    // Exemplu de apel
    n = 1221
    cout << "Numărul este palindrom: " << /* apel subprogram */ << endl;
    return 0;
}
```

7. Numar perfect

Scrie un subprogram care verifica daca un numar este perfect.

Exemplu input: n = 28

Exemplu output: Numarul este perfect: DA

Main (exemplu):

```
#include <iostream>
using namespace std;

// Subprogramul va fi implementat aici

int main() {
    // Exemplu de apel
    n = 28
    cout << "Numarul este perfect: " << /* apel subprogram */ << endl;
    return 0;
}
```

8. Putere a lui doi

Scrie un subprogram care verifica daca un numar este o putere a lui doi.

Exemplu input: n = 16

Exemplu output: Este putere a lui doi: DA

Main (exemplu):

```
#include <iostream>
using namespace std;

// Subprogramul va fi implementat aici

int main() {
    // Exemplu de apel
    n = 16
    cout << "Este putere a lui doi: " << /* apel subprogram */ << endl;
    return 0;
}
```

9. Inversul unui numar

Scrie un subprogram care returneaza inversul cifrelor unui numar.

Exemplu input: n = 4321

Exemplu output: Invers: 1234

Main (exemplu):

```
#include <iostream>
using namespace std;

// Subprogramul va fi implementat aici

int main() {
    // Exemplu de apel
    n = 4321
    cout << "Invers: " << /* apel subprogram */ << endl;
    return 0;
}
```

10. Suma cifrelor unui numar

Scrie un subprogram care returneaza suma cifrelor unui numar.

Exemplu input: n = 563

Exemplu output: Suma cifrelor: 14

Main (exemplu):

```
#include <iostream>
using namespace std;
```

```
// Subprogramul va fi implementat aici
```

```
int main() {
    // Exemplu de apel
    n = 563
    cout << "Suma cifrelor: " << /* apel subprogram */ << endl;
    return 0;
}
```