Solutii pentru exercitiile din sesiunea 1-1

1. Scrieti o functie care primeste ca si parametru un numar real, reprezentand temperatura in grade Fahrenheit si care intoarce rezultatul conversiei acesteia in grade celsius.

- Date de intrare: `63.5` (grade Fahrenheit)
- Date de iesire: `17.5` (grade celsius)
- Solutie:

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

float fahrenheitToCelsius(float gradeFahrenheit);

int main() {

    float gradeF = 40;
    cout << fahrenheitToCelsius(gradeF);

    return 0;
}

float fahrenheitToCelsius(float gradeFahrenheit) {
    float celsius = 5.0/9.0 * (gradeFahrenheit - 32);

    return celsius;
}
```

2. Scrieti un subprogram care primeste un numar natural `n` ca si parametru. Programul va intoarce oglinditul numarului `n`.

- Date de intrare: `12345`
- Date de iesire: `54321`
- Solutie:

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

int creeazaOglindit(int number);

int main() {

    int n = 12345;
    cout << creeazaOglindit(n);

    return 0;
```

```
}

int creeazaOglindit(int number) {
    int rezultat = 0;
    while (number > 0) {
        int ultimaCifra = number % 10;
        rezultat = rezultat * 10 + ultimaCifra;

        number = number / 10;
    }

    return rezultat;
}
```

3. Scrieti un subprogram care primeste un numar natural n ca si parametru. Subprogramul va intoarce 1 daca numarul este palindrom sau 0 in caz contrar

- Date de intrare: 34543
- Date de iesire: 1
- Date de intrare: 1212
- Date de iesire: 0
- Solutie:

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

int creeazaOglindit(int number);
int estePalindrom(int number);

int main() {

    int n = 1212;
    cout << estePalindrom(n);

    return 0;
}

int estePalindrom(int number) {
    int oglindit = creeazaOglindit(number);
    if (oglindit == number) {
        return 1;
    } else {
        return 0;
    }
}

int creeazaOglindit(int number) {
    int rezultat = 0;
    while (number > 0) {
        int ultimaCifra = number % 10;
        rezultat = rezultat * 10 + ultimaCifra;
```

```
        number = number / 10;
    }

    return rezultat;
}
```

4. Scrieţi definiţia completă a subprogramului numar, cu trei parametri, care primeşte prin intermediul parametrului n un număr natural format din cel mult 9 cifre, iar prin intermediul parametrilor c1 şi c2 câte o cifră nenulă. Subprogramul caută prima apariţie (de la stângab spre dreapta) a cifrei c1 în n, şi dacă aceasta apare, o înlocuieşte cu c2, iar următoarele cifre, dacă există, sunt înlocuite cu câte o cifră 0. Subprogramul furnizează tot prin n numărul astfel obţinut. Dacă cifra c1 nu apare în n, atunci valoarea lui n rămâne nemodificată.

- Date de intrare: n = 162448, c1 = 4 şi c2 = 7
- Date de iesire: 162708
- Solutie:

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

int numar(int n, int c1, int c2);


int main() {

    int n = 162448;
    int c1 = 4;
    int c2 = 7;
    cout << numar(n, c1,c2);

    return 0;
}

int numar(int n, int c1, int c2) {
    int rezultat = 0;
    int amGasitDejaC1 =0;
    int pozitie = 1;

    while (pozitie * 10 <= n) {
        pozitie = pozitie * 10;
    }

    while (pozitie != 0) {
        int primaCifra = n / pozitie;
        if (primaCifra == c1)
        {
            if (!amGasitDejaC1) {
                rezultat = rezultat * 10 + c2;
                amGasitDejaC1 = 1;
            } else {
```

```
                rezultat = rezultat * 10 + 0;
            }
        } else {
            rezultat = rezultat * 10 + primaCifra;
        }
        n = n % pozitie;

        pozitie = pozitie / 10;
    }

    return rezultat;
}
```

5. Funcţia `f` primeşte prin intermediul parametrului `n` un număr natural nenul ($2 \leq n \leq 200$), iar prin intermediul parametrului `a` un tablou unidimensional care conţine `n` valori întregi nenule (fiecare dintre aceste valori întregi având cel mult patru cifre). Funcţia returnează valoarea `-1` dacă numărul de valori negative din tabloul `a` este strict mai mare decât numărul de valori pozitive din tablou, valoarea 0 dacă numărul de valori negative din a este egal cu numărul de valori pozitive din tablou şi valoarea 1 dacă numărul de valori pozitive din tabloul `a` este strict mai mare decât numărul de valori negative din a. Scrieţi definiţia completă a funcţiei `f`.

- Date de intrare: `n = 8`, `a = {1, -1, 2, -3,-4,-5,-33,2}`
- Date de iesire: `-1`
- Date de intrare: `n = 8`, `a = {1, 1, 2, 3,-4,-5,-33,2}`
- Date de iesire: `1`
- Date de intrare: `n = 8`, `a = {1, 1, 2, -3,-4,-5,-33,2}`
- Date de iesire: `0`
- Solutie:

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

int f(int n, int a[]);

int main() {

    int n = 8;
    int a[] =  {1, 1, 2, 3,-4,-5,-33,2};

    cout << f(n, a);

    return 0;
}

int f(int n, int a[]) {
    int numarNegative = 0;
    int numarPozitive = 0;
    for (int i = 0; i < n; i++) {
        if (a[i] >= 0) {
```

```
                numarPozitive++;
            } else {
                numarNegative++;
            }
        }
        if (numarNegative > numarPozitive) {
            return -1;
        } else if (numarNegative < numarPozitive) {
            return 1;
        } else {
            return 0;
        }
    }
```