

Portofoliu modul 1

Chende-Roman Ana

Cuprins

| | |
|---------------------------|---|
| Cuprins | 1 |
| Enunt subiect 1 | 2 |
| Rezolvare subiect 1 | 2 |
| Enunt subiect 3..... | 4 |
| Rezolvare subiect 3 | 4 |
| Enunt subiect 8..... | 6 |
| Rezolvare subiect 8 | 6 |
| Enunt subiect 9..... | 8 |
| Rezolvare subiect 9 | 9 |

Enunt subject 1

Subiectul nr. 1:

- a) Subprogramul **cifre_pare** are ca parametru un număr natural **x** din intervalul $[0, 10^4]$. Subprogramul returnează suma cifrelor pare ale numărului **x** sau valoarea **-1**, dacă numărul nu conține nicio cifră pară. Scrieți la calculator definiția completă a subprogramului.
Exemplu: pentru **x=1234**, subprogramul returnează **6** iar pentru **x=9753**, subprogramul returnează **-1**.
- b) Fișierul **suma.in** conține pe prima linie un număr natural **n** ($1 \leq n \leq 100$), iar pe a doua linie un șir de **n** numere naturale din intervalul $[0, 10^4]$, separate prin câte un spațiu. Se cere scrierea și executarea la calculator a unui program care să calculeze suma cifrelor pare pentru fiecare număr de pe a doua linie a fișierului apoi să afișeze suma maximă calculată și primul număr care are această sumă, separate prin câte un spațiu. Programul va folosi apeluri utile ale subprogramului **cifre_pare**. Dacă nu există niciun număr care să conțină cifre pare, atunci pe ecran se afișează mesajul **doar cifre impare**.
Exemplu: dacă fișierul **suma.in** conține pe primul rând numărul **6**, iar pe al doilea rând numerele **2048 88 79 1848 1379 8606** se va afișa **20 1848**.

Rezolvare subiect 1

```
#include <iostream>

#include <fstream>

using namespace std;

int cifre_pare(int x) {
    //cout << "START " << x << endl;

    int uc, s = 0, are_c_p = 0;

    while(x > 0) {
        uc = x%10;

        x = x/10;

        if(uc%2 == 0) {
            are_c_p = 1;

            s = s + uc;

            //cout << s << " _ " << uc << endl;
        }
    }

    if(are_c_p == 1){
```

```

        return s;
    } else {
        return -1;
    }
}

int main() {
    ifstream f("suma.in");

    int n; //nr de pe prima linie din fisier

    int v[100]; //vectorul care va contine cele n numere de pe linia 2 din fisier

    int i;

    int s_max = -1;

    int s_c_p;

    int nr_s_max;

    f >> n;

    cout << "n: " << n << endl;

    for(i = 0; i < n; i++) {
        f >> v[i];
    }

    f.close();

    for(i = 0; i < n; i++) {
        s_c_p = cifre_pare(v[i]);

        cout << v[i] << ", s = " << s_c_p << endl;

        if(s_c_p > s_max) {
            s_max = s_c_p;

            nr_s_max = v[i];
        }
    }

    cout << "suma maxima: " << s_max << ", pentru numarul: " << nr_s_max;

    return 0;
}

```

Enunt subject 3

Subiectul nr. 3:

a) Subprogramul **gaseste_cifra** are doi parametri:

- **x**, prin care primește un număr natural din intervalul $[0, 10^9]$;
- **c**, prin care primește o cifră.

Subprogramul returnează numărul de apariții ale cifrei **c** în scrierea numărului **x**. Scrieți la calculator definiția completă a subprogramului.

Exemplu: pentru **x=1323** și **c=3**, subprogramul returnează **2**; pentru **x=1323** și **c=4**, subprogramul returnează **0**.

b) Se citește de la tastatură un număr natural **nr** din intervalul $[0, 10^9]$. Se cere scrierea și executarea la calculator a unui program care să calculeze și să scrie, în fișierul **control.out**, valoarea formată din numărul de apariții ale fiecărei cifre în scrierea numărului **nr**, folosind apeluri utile ale subprogramului **gaseste_cifra**, iar pe linia următoare în fișier se va scrie cifra de control a lui **nr**. Cifra de control este cifra obținută prin însumarea cifrelor numărului și repetarea procedurii pentru suma obținută, până când se obține o singură cifră.

Exemplu: pentru **nr=23242527**, fișierul **control.out** va conține:

- pe primul rând, numărul **41110100** (pentru că cifrele **0** și **1** nu apar, **2** apare de **4** ori, **3** apare o dată, **4** apare o dată, **5** apare o dată, **6** nu apare, **7** apare o dată, **8** și **9** nu apar).
- pe al doilea rând, numărul **9** ($2+2+2+2+3+4+5+7=27$; $2+7=9$).

Rezolvare subiect 3

```
#include <iostream>
```

```
using namespace std;
```

```
int gaseste_cifre(int x, int c) {  
    int contor = 0;  
    int uc;  
    while (x > 0){  
        uc = x % 10;  
        x = x / 10;  
        //cout << "c = " << uc << endl;  
        if (uc == c) {
```

```

        contor += 1;
    }
}
return contor;
}

```

```

int main() {
    int n, uc, cnt, c;
    int nr_ap_cifre = 0;
    int s = 0, aux;
    cout << "n = ";
    cin >> n;
    for (c = 0; c <= 9; c++) {
        cnt = gaseste_cifre(n, c);

        nr_ap_cifre = nr_ap_cifre * 10 + cnt;
    }
    cout << nr_ap_cifre << endl;

    // calculare suma cifre numar
    while (n > 0){
        uc = n % 10;
        n = n / 10;
        s = s + uc;
    }

    //cat timp suma nu este de o singura cifra, voi face suma cifrelor
    //pana cand suma devine o cifra
    while (s > 10) {
        aux = s;
        s = 0;
        while (aux > 0) {

```

```

        uc = aux % 10;

        aux = aux / 10;

        s = s * 10 + uc;
    }
}

cout << s;

return 0;
}

```

Enunt subiect 8

Subiectul nr. 8:

- a) Subprogramul **nr_prim** are un singur parametru **n** prin care primește un număr natural cu cel mult patru cifre și returnează **1** dacă **n** este prim sau **0** altfel.
 Subprogramul **cmmdc** are doi parametri prin care primește câte un număr natural nenul cu cel mult patru cifre. Subprogramul returnează cel mai mare divizor comun al celor două numere.
 Scrieți la calculator definițiile complete ale celor două subprograme.

5



MINISTERUL EDUCAȚIEI NAȚIONALE



INSPECTORATUL ȘCOLAR AL
MUNICIPIULUI BUCUREȘTI

- b) Fișierul **numere.in** conține pe prima linie cel puțin două și cel mult 10^6 numere naturale din intervalul $[1, 10^4]$, separate prin câte un spațiu. Se cere scrierea și executarea la calculator a unui program care să afișeze pe ecran numărul perechilor de numere citite consecutiv din fișier care nu sunt numere prime, dar sunt numere prime între ele.

Exemplu: dacă fișierul **numere.in** conține numerele:

35 45 22 11 8 35 256 1000

atunci se afișează **3**

Explicație: Se numără perechile **(45, 22)**, **(8, 35)**, **(35, 256)**

Rezolvare subiect 8

```
#include <iostream>
```

```

#include <fstream>

using namespace std;

int nr_prim(int nr) {
    int este_prim = 1;
    int i;
    for(i = 2; i <= nr/2 && este_prim == 1; i++) {
        if(nr%i == 0) {
            este_prim = 0;
        }
    }

    return este_prim;
}

int cmmdc(int nr1, int nr2) {
    int rest;
    //cout << nr1 << ", " << nr2;
    while(nr2 != 0) {
        rest = nr1 % nr2;
        nr1 = nr2;
        nr2 = rest;
    }
    //cout << ", cmmdc = " << nr1 << endl;
    return nr1; //cand nr2 ajunge 0, nr1 va fi cmmdc
                //daca nr1 (cmmdc) este 1, numerele sunt prime intre ele
}

int main() {
    ifstream f("numere.in");
    int contor = 0;

```

```

int a, b; //in a voi citi un un numar din fisier, in b numarul urmator
f >> a;
while(f >> b) { // cat timp am numere in fisier, cand nu mai sunt, se iese din bucla
    if(cmmdc(a, b) == 1 && nr_prim(a) == 0 && nr_prim(b) == 0) {
        cout << "nu sunt prime dar sunt prime intre ele: (" << a << "," << b << ")" <<
endl;

        contor++;
    }
    a = b; // a devine al doilea numar citit,
        // in while in b se va pune al treilea numar
}
cout << contor;
f.close();
return 0;
}

```

Enunt subject 9

Subiectul nr. 9:

- a) Oglinditul unui număr este numărul scris cu cifrele primului număr, în ordine inversă. Subprogramul **oglinnda** primește ca parametru un număr natural **x** din intervalul $[1, 10^6]$ și returnează oglinditul acestuia. Scrieți la calculator definiția completă a subprogramului.

Exemplu: pentru **x=12401**, subprogramul returnează numărul **10421**.

- b) Se citește de la tastatură două numere naturale **a** și **b** din intervalul $[1, 10^6]$. Un număr este palindrom dacă este egal cu oglinditul său. Se cere scrierea și executarea la calculator a unui program care scrie în fișierul **palindrom.out**, separate prin câte un spațiu, în ordine crescătoare, toate numerele din intervalul $[a, b]$ care sunt palindromuri. Apoi, programul citește de la tastatură un număr natural **k** din intervalul $[1, 10^7]$ și scrie, pe următorul rând al fișierului **palindrom.out**, palindromul asociat lui **k**. Palindromul asociat unui număr este cel mai mic palindrom mai mare decât numărul inițial. Dacă numărul **k** este palindrom, se va scrie mesajul **palindrom**.

Pentru determinarea valorilor cerute, programul va folosi apeluri utile ale subprogramului **oglinnda**.

Exemplu:

- pentru **a=11** și **b=130**, fișierul **palindrom.out** va conține pe primul rând numerele **11 22 33 44 55 66 77 88 99 101 111 121**
- pentru **k=39995**, fișierul **palindrom.out** va conține pe al doilea rând numărul **40004**

Subiectul nr. 10:

Rezolvare subiect 9

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

int oglinda(int numar) {
    //construim oglinditul luand cifra cu cifra numerele de la sfarsit spre inceput
    int oglindit = 0;
    int uc;
    while(numar != 0) {
        uc = numar % 10;
        oglindit = oglindit*10 + uc;
        numar = numar/10;
    }
    return oglindit;
}

int main() {
    int a, b, i, k;
    ofstream fout("palindrom.out"); // fout - variabila asociata cu fisierul palindrom.out

    cout << "dati primul numar:   a = ";
    cin >> a;
    cout << "dati al doilea numar: b = ";
    cin >> b;
    cout << "palindroame in intervalul [" << a << ", " << b << "]" << endl;
    for(i = a; i <= b; i++) {
        if(i == oglinda(i)) {
            cout << i << " ";
        }
    }
}
```

```

        fout << i << " ";

    }

}

fout << endl;


cout << endl << "dati numarul k: ";
cin >> k;


if(oglinda(k) == k) {
    cout << "palindrom";
    fout << "palindrom";
} else {
    k++;
    while(oglinda(k) != k) {
        k++;                // il tot maresc pe k pana am un palindrom
    }
    cout << k;
    fout << k;
}


fout.close(); //inchidem fisierul
return 0;
}

```