

Proiect de lecție / unitate de învățare

Unitatea de învățământ: Colegiul Național Petru Rareș Suceava

Disciplina: Informatică

Clasa: a X-a intensiv informatică

Data¹: _____

Profesor: Tiberiu Socaciu

Unitatea de învățare: Stringuri

Tema lecției: Stringuri. Maximum Value of a String in an Array

Tipul lecției: practică, de laborator

Scopul lecției: sistematizare și consolidare abilități și deprinderi practice

Competențe generale:

- a) **CG1:** Elaborarea algoritmilor de rezolvare a problemelor
- b) **CG2:** Implementarea algoritmilor într-un limbaj de programare

Competențe specifice:

- a) **CS1:** Analiza problemei în scopul identificării metodei de programare adecvate pentru rezolvarea problemei
- b) **CS2:** Aplicarea creativă a metodelor de programare pentru rezolvarea unor probleme intradisciplinare sau interdisciplinare sau a unor probleme cu aplicabilitate practică
- c) **CS3:** Utilizarea tehnicilor moderne în implementarea aplicațiilor

Obiective operaționale². La finalul lecției, elevul poate:

- a) **O1:** să demonstreze posibilitatea de a folosi corect noțiunile prezentate
- b) **O2:** să aplice noțiunile prezentate într-un caz concret
- c) **O3 :** să analizeze relațiile dintre funcțiile programului
- d) **O4:** să descrie în limbaj științific mecanismele de interdependență dintre modulele programului
- e) **O5:** să formuleze propria opțiune pentru o decizie optimă

Strategii didactice:

- a) **Mijloace de învățământ:** calculatoarele din laboratoarele de informatică, directorul FTP_Elevi -> Materiale_Profesorii prezent pe toate cele 30 stații de lucru, tableta grafică precum și resursele incluse în prezentul plan de lecție³ respectiv:

Link pentru enunțul și tema lecției	TBC ⁴
Link pentru mind map-ul lecției ⁵	TBC ⁶
Link pentru produsul final al lecției ⁷	TBC ⁸
Link și cod QR pentru activitatea independentă ⁹ și pentru tema acasă a elevilor	TBC ¹⁰
Generator de numere aleatoare	https://g.co/kgs/s3eGCQ4

- b) **Metode de învățământ:** expunerea, algoritimizarea, demonstrația, modelarea, problematizarea, conversația euristică, descoperirea sau brainstormingul.
- c) **Forme de activitate:** individuală, frontală, lucrul pe grupe pentru orele de laborator, lucrul pe grupe mici, lucrul în perechi.
- d) **Metode de evaluare:** evaluare orală, evaluare scrisă, observarea sistematică a elevilor sau prin teste și/sau fișe de evaluare.

Bibliografie:

- a) cursul *Make Technology your Friend*
- b) site-ul LeetCode, www.leetcode.ro
- c) Eusebiu Toader, Tiberiu Socaciu, C++ și Python, teme de interviu rezolvate, MatrixRom, 2023, <https://www.matrixrom.ro/produs/c-si-python-teme-de-interviu-rezolvate/>

¹ Se va completa, după caz

² rezultă din derivarea competențelor specifice

³ link-ul lecției va fi disponibil elevilor și înafara orelor de curs pentru ca aceștia să poată accesa oricând materialele folosite

⁴ Se va completa ulterior

⁵ plan de lecție alternativ, necesar pentru susținerea activităților A și B din ”dirijarea învățării”

⁶ Se va completa ulterior

⁷ activitatea C

⁸ Se va completa ulterior

⁹ în clasă

¹⁰ Se va completa ulterior

Succesiunea activităților de instruire

Nr. crt.	Evenimentul didactic	Durata (min.)	Obiective operaționale	Activitatea de instruire		Strategia didactică			
				Activitatea profesorului	Activitatea elevilor	Mijloace de învățământ	Metode și procedee	Forme de organizare	Forme de evaluare
1.	Moment organizatoric Prezentarea temei abordate și a obiectivelor lecției Captarea atenției	2		Salutul, efectuarea prezenței, verificarea funcționării rețelei de calculatoare, a FTP-ului. Prezentarea temei și a obiectivelor lecției.		AdServio	Conversația euristică	Frontal	Observarea sistematică a elevilor
2.	Dirijarea învățării	10	O1	Activitatea A	Notează lecția pe caiet sau calculator, adresează întrebări	Videoproiector	Învățare activă, prin cooperare	Frontal	
		10	O2	Activitatea B	Notează lecția pe caiet sau calculator, adresează întrebări	Videoproiector	Învățare activă, prin cooperare	Individual Frontal	Observarea sistematică a elevilor
		10	O3	Activitatea C	Notează lecția pe caiet sau calculator, adresează întrebări	Videoproiector	Învățare activă, prin cooperare	Frontal	
3	Transferul cunoștințelor/ consolidarea cunoștințelor.	10	O4	Problematizare legată de relația dintre funcțiile programului. Se invită un elev la tablă pentru a realiza un modul necesar rezolvării problemei	Implementează funcția pe calculator	Rețeaua de calculatoare Videoproiector	Algoritmizarea	Individual Frontal	Orala
4	Fixarea cunoștințelor și asigurarea feedback-ului	7	O5				Problematizare a	Frontal	Scrisa
5	Tema pentru acasă	1	O1, O2, O3, O4, O5	Solicită elevilor să completeze ca temă pentru acasă o problema de pe LeetCode, aleasa aleator, împreună Generează sarcina de lucru pentru acasă în Classroom	Notează tema de casă pe caiet sau calculator	Generator de numere aleatoare Google Classroom			Scrisa

Enunt

Valoarea unui șir alfanumeric poate fi definită ca:

- Reprezentarea numerică a șirului în baza 10, dacă este format numai din cifre.
- Lungimea șirului, în rest.

Dându-se o matrice șiruri de șiruri alfanumerice, returnează valoarea maximă a oricărui șir în șiruri.

Exemplu 1:

Intrare: strs = ["alic3","bob","3","4","00000"]

Ieșire: 5

Explicație:

- „alic3” constă atât din litere, cât și din cifre, deci valoarea sa este lungimea sa, adică 5.
 - „bob” este format doar din litere, deci valoarea sa este și lungimea, adică 3.
 - „3” este format numai din cifre, deci valoarea sa este echivalentul numeric, adică 3.
 - „4” constă, de asemenea, numai din cifre, deci valoarea sa este 4.
 - „00000” este format numai din cifre, deci valoarea sa este 0.
- Prin urmare, valoarea maximă este 5, a lui „alic3”.

Exemplu 2:

Intrare: strs = ["1","01","001","0001"]

Ieșire: 1

Explicație:

Fiecare șir din matrice are valoarea 1. Prin urmare, returnăm 1.

Constrângeri:

- $1 \leq \text{strs.lungime} \leq 100$
- $1 \leq \text{strs}[i].\text{lungime} \leq 9$
- $\text{strs}[i]$ constă numai din litere și cifre engleze mici.

Rezolvare. Prima solutie

```
// # include <string.h> - ar trebui sa o folosim DOAR daca folosim functiile
// de tip string; noi, cat timp DOAR citim un ASCIIZ cu cin >>, nu trebuie sa
// folosim aceasta biblioteca,
// caci functia
// std::istream& operator>>(std::istream&, char*)
// este DEJA definita in iostream
// totusi, am un strlen in functia de mai jos...
# include <string.h>

// subproblema 1
// am un sir de caractere -> sa vad daca TOATE caracterele sunt cifre
// verif_cifre ("001010101") = 1
// verif_cifre ("alic3")      = 0
int verif_cifre(char sir_de_caractere[9 + 1]) // aici am vrut sa pun
dimensiunea !!!
{
for (int i = 0; i < strlen(sir_de_caractere); i++)
    // if (not ('0' <= sir_de_caractere[i] and sir_de_caractere <= '9'))
    // aplic de morgan
    // if (not '0' <= sir_de_caractere[i] or not sir_de_caractere <= '9'))
    // aplic negatia
    if ('0' > sir_de_caractere[i] or sir_de_caractere[i] > '9')
        return 0;
```

```

return 1;
}

# include <iostream> // pentru cout si endl
// using namespace std; daca vreau sa renunt la namespace-ul std urmat de
operatorul rezolutiv ::

// subproblema 2
// am un sir de caractere care sunt TOATE cifre, vreau valoarea numarului
// valoare_numar("101011") = 101011
// schema lui horner
int valoare_numar(char sir_de_caractere[]) // observati ca nu am pus
dimensiunea!!!
{
int suma = 0;
for (int i = 0; i < strlen(sir_de_caractere); i++)
    suma = suma * 10 + (sir_de_caractere[i] - '0');
return suma;
}

// rezolvarea problemei
int calcul(int n, char vector_de_sir_de_caractere[][9 + 1]) // atentie la acel
9 + 1
{
int maximum = -1; // o santinela care va putea trece si de testul cu primul
candidat la maxim
for (int i = 0; i < n; i++)
    if (verif_cifre(vector_de_sir_de_caractere[i])) // daca esueaza acest test,
eu ma opresc, caci, oricum, nu are sens sa calculam maximum
    {
        int valoarea_numarului = valoare_numar(vector_de_sir_de_caractere[i]);
        if (valoarea_numarului > maximum)
            maximum = valoarea_numarului;
    }
    else
        return n;
return maximum;
}

// citirea datelor
// n parametru de tip referinta ca sa poata fi modificat in interiorul
functiei
// alternativa ar fi fost tehnica din C cu pointer
// de ce nu pun operator de dereferentiere implicita la
vector_de_sir_de_caractere?
// pentru ca el este un tablou de 100 de elemente, adica el este un pointer
void citirea_datelor(int &n, char vector_de_sir_de_caractere[][9 + 1]) //
atentie la acel 9 + 1!!!
{
std::cin >> n;
for (int i = 0; i < n; i++)
    // vector_de_sir_de_caractere[i] este un vector de 9 + 1 char, adica el este
un pointer la un tablou
    // de char in format ASCII
    std::cin >> vector_de_sir_de_caractere[i];
}

int main()
{
int n;

```

```
// max 100 cuvinte, de lungime 9; dar cuvintele le citesc cu string.h
(asciiiz), deci, am nevoie de \0 la final
char strsr[100][9 + 1]; // calculul 9 + 1 se va face la compilare, pentru a
putea genera codul care realizeaza alocarea
citirea_datelor(n, strsr);
std::cout << calcul(n, strsr) << std::endl;
return 0;
}
```