

Informatii utile pentru examenul de bacalaureat

- Recomand a se scrie de mana fiecare dintre definitiile pentru topic-urile de mai jos.
- Recomand a se desena de fiecare data atunci cand invatam un topic nou, daca se poate (de exemplu la grafuri)

Pseudocod

- Aici gasim efectiv tot ce este nevoie pentru a intelege pseudocod:
<https://www.pbinfo.ro/articole/23972/limbajul-pseudocod>
 - Recomand a se parcurge cu atentie toate structurile repetitive
 - Trebuie stiut cum se poate inlocui fiecare dintre structurile repetitive cu o alta structura repetitiva:
 - De exemplu:
 - Din `pentru... executa in executa .. cat timp`
 - ATENTIE foarte mare la diferenta intre `repete...pana cand si executa...cat timp`

Grafuri

- Printre conceptele care trebuiesc stapanite avem:
 - Ce inseamna un graf
 - Ce inseamna un varf
 - Ce inseamna o muchie
 - Ce inseamna un graf orientat
 - Ce inseamna un graf neorientat
 - Ce inseamna un graf complet
 - Cate muchii are un graf complet
 - Gradul unui varf
 - Grad intern si grad extern pentru cele orientate
 - Matricea de adiacenta
 - Liste de adiacenta
 - Graf partial
 - Subgraf
 - Graf complementar
 - Graf nul
 - Graf complet
 - Graf bipartit, bipartit complet
 - Graf regulat
 - Lant
 - simplu
 - elementar
 - compus
 - Ciclu
 - elementar

- lungimea ciclului
- Circuit
- Diferenta intre
 - Drum
 - Lant
 - Ciclu
 - Circuit
- Graf conex
- Componente conexe
- Arbore, Padure
- Graf hamiltonian / ciclu hamiltonian
- Graf eulerian / ciclu eulerian
- Diferite formule (numarul total de muchii, etc), teoreme,
 - Totul poate fi gasit in link-urile de mai jos
- Vector de tati
- Ce este o frunza
- Descendenti, descendenti directi, ascendenti, ascendenti directi

Grafuri neorientate

- Materialul complet se gaseste aici: <https://www.pbinfo.ro/articole/810/grafuri-neorientate>

Grafuri orientate

- Materialul complet se gaseste aici: <https://www.pbinfo.ro/articole/509/grafuri-orientate>

Structuri

- Ce este o structura
- Cum accesam membrii unei structuri
- Cum definim o variabila de tipul unei structuri
- Repetare caz in care putem folosi ca si membru intr-o structura, o proprietate de tip `struct`
 - De exemplu:

```
struct data_calendaristica {  
    int zi;  
    int luna;  
    int an;  
};  
  
struct student {  
    char nume[50];  
    char prenume[50];  
    data_calendaristica data_nasterii;  
};
```

Functii / Subprograme

- Pentru fiecare din tipurile de functii de mai jos, sa stim sa le definim corect:
 - Functii care intorc valori `int sum(int a, int b)`
 - Functii care nu intorc (`void`) `void afiseaza(int a)`
 - Functii care intorc prin parametri `void faCeva(int n, int& out)`
 - Functii recursive

Functii pentru prelucrarea sirurilor de caractere

- Functii cruciale:
 - strcpy
 - strcat
 - strtok
 - strncpy
 - strchr
 - strstr
- Pentru fiecare, recomand a se scrie un exemplu in care sa se observe utilitatea functiei respective
- Mai multe detalii gasim aici: <https://cplusplus.com/reference/cstring/>

Vectori/Array-uri

- vector de frecventa
- eliminare elemente din vector

Matrici

- Formula diagonala principala
- Formula diagonala secundara
- Elementele de deasupra si de dedesubtul celor 2 diagonale

Formule matematice/diverse

- Distanta intre doua puncte
- Numar prim
- Palindrom
- Cum construim inversul unui numar (e.g 321 -> 123)
- Cum eliminam cifrele dintr-un numar

Fisiere

- Citire din fisier

- Scriere in fisier
- Adaugare in fisier
 - Aici vreau sa evidentiez ca este posibil ca atunci cand scriem intr-un fisier, putem sa adaugam la ceea ce deja exista in fisier. Si acest lucru se face astfel:

```
ofstream fout("bac.txt", ios::app);
```

- Daca nu se specifica parametrul `ios::app`, la scriere, se va suprascrie continutul deja existent cu continutul pe care vrem sa il adaugam.
- Inchiderea fisierelor