

Teme și aplicații

Formatori:

Tutor: [Stângaciu Valentin](#)  

Tutor: [Belu Claudiu-Marcel](#)  

+3

Data de începere a cursului:

 25.09.2023

 [Utilizatori înscriși](#)

 [Calendar](#)

 [Note](#)

 [Cursurile mele](#) [S1-L-AC-CTIRO1-PC](#) [Laborator 11: Tipuri de date definite de utilizator](#) [Teme și aplicații](#)

Teme și aplicații

1. Să se scrie un program ce definește un tip de date utilizator ce modelează un număr complex și o serie de funcții ce realizează operațiile de bază cu numere complexe. Programul se va testa prin numere introduse de la tastatură
 - o funcție de adunare ce primește ca argument două numere complexe și returnează un număr complex ce reprezintă suma celor 2 numere complexe primite ca și argumente
 - o funcție de scădere ce primește ca argument două numere complexe și returnează un număr complex ce reprezintă diferența celor 2 numere complexe primite ca și argumente
 - o funcție de înmulțire ce primește ca argument două numere complexe și returnează un număr complex ce reprezintă produsul celor 2 numere complexe primite ca și argumente
 - o funcție de afișare a unui număr complex în formă algebrică: $a+bj$ - primește ca argument numărul complex
 - o funcție de calcul a modulului unui număr complex - primește ca argument numărul complex și returnează un double ce reprezintă modulul numărului complex. Se va folosi funcția `sqrt` din `<math.h>`. Pentru compilare se adaugă parametrul `-lm` la gcc.
2. O structură conține ora (ca și o altă structură potrivită) la care s-a măsurat o anumită temperatură și valoarea acestei temperaturi (real). Se cere $n \leq 10$ și apoi n temperaturi cu ora la care au fost măsurate. Se cere apoi o oră de început și una de sfârșit. Să se afișeze media temperaturilor care au fost măsurate în acel interval orar, inclusiv în capetele acestuia.
3. Se consideră o structură *Persoana* care are un câmp *nume* și altul *varsta*. Să se scrie o funcție care primește ca parametru o persoană și îi modifică numele astfel încât prima literă să fie mare iar restul mici. Să se testeze funcția cu o persoană citită de la tastatură.
4. Se citește un n oricât de mare și apoi n puncte în plan, definite prin coordonatele lor (x,y) . Să se afișeze toate punctele, grupate în seturi de puncte care sunt pe aceeași linie orizontală (au același y). Memoria folosită va fi minimă.
5. Să se implementeze o bază de date ce modelează animale. Animalele pot fi insecte, păsări, mamifere și pești. O insectă se caracterizează prin număr picioare și durata medie de viață în zile, o pasăre prin viteza de zbor și lungimea aripilor, un mamifer prin greutate, înălțime și tip de mâncare (carnivor, ierbivor, omnivor) iar un pește prin greutate, adâncimea maximă de înot și salinitatea apei (în procentaj). Să se definească structurile de date necesare cât mai abstract posibil și cu amprenta de memorie minimă. Se va defini un tablou de astfel de structuri și se vor implementa funcții de adăugare animal, ștergere animal și printare animale.
6. Se citește de la intrarea standard un număr necunoscut de perechi de 2 numere despărțite prin `.` - punct (exemplu 34.21 17.88 90.100). Aceste perechi reprezintă coordonatele unor puncte în sistemul cartezian primul reprezentând abscisa (x) și al doilea ordonata (y). Se va proiecta și implementa o structură de date corespunzătoare și se vor stoca toate punctele citite într-un tablou. Nu se cunoaște numărul de perechi de numere astfel ca se va alocă dinamic tabloul și se va folosi minimul de memorie. Programul va afișa la ieșirea standard punctul cel mai apropiat de origine.
7. Să se implementeze o bază de date care conține următoarele tipuri de produse:
 - stick USB: interfata (A sau C), capacitate
 - baterie: voltaj
 Fiecare produs va avea asociate un pret, stoc și un cod unic. Operațiile vor fi:
 - introducere - introduce un nou produs, inclusiv codul sau. Dacă codul există deja, se incrementează stocul, altfel se consideră stoc=1
 - listare - se listează doar produsele existente pe stoc
 - ștergere - se dă un cod și se decrementează stocul. Produsele cu stoc 0 rămân în baza de date. Dacă deja este 0, se emite mesaj de atenționare.
 Programul va folosi alocare dinamică, astfel încât să se folosească doar strictul necesar de memorie. Utilizatorul va avea la dispoziție un meniu, de unde va putea apela fiecare funcție.