**Fişa de laborator**

**Fundamentele programării**

|  |  |  |
| --- | --- | --- |
| **Laborator 1. Generalităţi** |  |  |
| Familiarizarea cu mediul de programare. Prezentarea elementelor structurale a programelor în C#. Efectele principalelor noţiuni întȃlnite în literatura de specialitate: Editare, Rulare, Compilare, „Design Mode”, „Run time”.  Stabilirea tipului de proiect, a setărilor acestuia şi a modului general de implementare şi identificare a programelor realizate. |  |  |
| **Laborator 2. Elemente structurale ale programelor** |  |  |
| Identificarea tipurilor de date standard şi a intervalelor de definiţie a acestora.  Identificarea principalelor clase de operatori şi realizarea de programe demonstrative cu acţiunea acestora în contextul diferitelor tipuri de date şi a precedenţei operatorilor.  Efectuarea de exemple pentru evidetierea importantei cunoasterii precedentei operatorilor, cat si a modului de executie functie de tipul de date ale operanzilor utilizati.  Familiarizarea cu unele funcţii predefinite.   * Probleme a căror rezolvare constă în implementarea unor formule matematice  1. Să se construiască un algoritm care calculează aria unui triunghi pentru care se cunosc laturile acestuia 2. Construiţi un algoritm care transformă o valoare dată în grade celsius în valoarea corespunzătoare în grade Farenheid |  |  |
| **Laborator 3. Algoritmi elementari (Probleme de decizie)** |  |  |
| * Instrucţiuni de decizie. Combinarea operatorilor relaţionari şi logici.  1. Minimul dintre 2,3, şi 4 numere citite de la tastatură 2. Se dau 3,4 sau 5 numere întregi să se detemine dacă există 2 sau 3 valori identice 3. Construiţi un algoritm pentru rezovarea ecuaţiei de grad II 4. Calculul în diferite puncte pentru unele funcţii matematice definite pe porţiuni   [Ex. calculaţi ]   1. Se dau 3 numere întregi să se afişeza în ordine crescătoare |  |  |
| **Laborator 4. Algoritmi elementari (Probleme de repetiţie)** |  |  |
| * Instrucţiuni repetitive (for, do{}while şi while)  1. Ridicarea la putere a unui număr (iterativ) 2. Calculul factorialului (iterativ) 3. Calculul sumei primelor n numere naturale (iterativ şi comaparea rezultatelor cu formula lui Gauss) 4. Test de primalitate pe baza calculului numărului de divizori 5. Transformări de baze |  |  |
| **Laborator 5. Algoritmi elementari (Probleme de repetiţie)** |  |  |
| * Clasa de probleme a căror rezolvare constă în procesarea cifrelor unui număr  1. Suma cifrelor unui număr 2. Minimul cifrelor unui număr 3. Sa se genereze oglinditul unui numar 4. Algoritm de determinare dacă un număr este palindrom 5. Algoritm de determinare dacă un număr are cifrele în ordine crescătoare / descrescătoare |  |  |
| **Laborator 6. Algoritmi elementari (şiruri (liste) de numere)** |  |  |
| * Clasa de probleme în care datele de intrare se introduc secvenţial şi se procesează în momentul introducerii. Se introduc numere de la tastatură, pȃnă la valoarea (-1). Afişaţi:  1. Suma numerelor 2. Minimul/Maximul acestora 3. Numărul de numere prime 4. Media aritmetică 5. Determinarea numarului de aparitii a unui caracter intr-un text introdus de la tastarura |  |  |
| **Laborator 7. Siruri de caractere** |  |  |
| * Procesarea şirurilor de caractere (Tipul de dată „string”, citire, afişare)  1. Construirea algoritmică a funcţiilor definite pe şiruri de caractere din C/C++ (strcpy, strcat, strlen,strncat ... ), adaptate pentru variabile de tip string din C#.  * Algoritmi de transformare  1. Crearea si afisarea unui sir de caractere format din caractere introduse de la tastatura unul cate unul 2. Algoritm de transformare a unui şir de caractere în număr întreg 3. Algoritm de transformare a unui şir de caractere în număr real |  |  |
| **Laborator 8. Vectori** |  |  |
| 1. Norma, suma şi produsul elementelor unui vector 2. Minmul sau maximul elementelor unui vector 3. Soratrea prin selecţie 4. Sortarea prin selectie directă 5. Sortarea prin metoda bulelor 6. Sortarea prin insertie |  |  |
| **Laborator 9. Matrici** |  |  |
| 1. Implementarea algortimului lui Sarus pentru calculul determinantului unei matrici de 3x3 2. Suma a două matrici 3. Identificarea diagonalelor într-o matrice şi calculul urmei unei matrici 4. Transpusa unei matrici |  |  |
| **Laborator 10. Funcţii. Transmiterea parametrilor la funcţii** |  |  |
| * Construirea şi implementarea unor funcţii * Programe demonstrative care arată modul de transmitere a parametrilor unei funcţii  1. Determinarea numarului de numere prime introduse de la tastatura cu ajutorul unei functii care determina daca un numar este prim 2. Rezolvarea problemei ce presupune calculul sumei a doua matrici, dar utilizandu-se functii pentru citirea, scrierea si adunarea matricilor |  |  |
| **Laborator 11. Tipuri de date definite de utilizator** |  |  |
| 1. Construirea tipului de dată număr complex şi implementarea operatorilor de adunare, normă, înmulţire a acestor numere 2. Construirea tipului de dată număr raţional şi implementarea principalilor operatori pentru aceste numere. |  |  |
| **Laborator 12. Recursivitate** |  |  |
| 1. Funcţia factorial 2. Generarea şirului lui Fibonacci 3. Transformarea unor algoritmi iterativi în algoritmi recursivi (Bubble Sort) 4. Simularea instructiunii FOR cu ajutorul unei functii recursive 5. Generarea indicilor unui tablou multidimensionali cu ajutorul unei functii recursive |  |  |
| **Laborator 13. Fişiere** |  |  |
| Citirea şi salvarea datelor în fişiere de tip text.   1. Realizarea unui program care citeşte date întregi dintr-un fişier, reprezentȃnd un vector de dimensiune n 2. Copierea datelor dintr-un fişier în altul 3. Adăugarea datelor dintr-un fişier în altul. 4. Sa se determine numarul de aparitii a caracterelor alfabetului englez intr-un fisier, fara a tine cont daca sunt litere mici sau mari 5. Sa se determine de cate ori apare un sir dat intr-un fisier |  |  |
| **Laborator 14. variabile aleatoare** |  |  |
| 1. Generarea unei permutări a unei mulţimi 2. Implementarea algoritmului jocului „giceşte numărul” 3. Generarea unor numere cu diferite restricţii (cifre distincte) 4. Generarea unor serii de numere cu diferite restricţii (numere crescătoare într-un anumit interval) |  |  |

Algoritmii dezvoltaţi pentru rezolvarea problemelor de la 1 pȃnă la 54 inclusiv, optimizările, adnotările şi rezultatele experimentale obţinute pe baza acestora vor constitui portofoliul fiecărui student.