Tema lab.5

NICOLETA RADU

# Exercitiul 1

Se consideră două tablouri unidimensionale A și B: A=(1,7,9) și B=(5,7,8,12).

Determinaţi elementele unui tablou obținut prin interclasarea tablourilor A și B.

Scrieţi elementele tabloului determinat în ordinea din acesta, separate prin exact un spaţiu.

void printArray(int arr[], int size)

{

int i;

for (i = 0; i < size; i++)

cout << arr[i] << " ";

cout << endl;

}

int main()

{

int tablou1[] = {1,7,9};

int tablou2[] = {5,7,8,12};

printArray(tablou1,3);

printArray(tablou2,4);

const int size1 = sizeof(tablou1) / sizeof(tablou1[0]);

const int size2 = sizeof(tablou2) / sizeof(tablou2[0]);

const int finalSize = size1 + size2;

cout << "Marimea celor doua tablouri este: " << finalSize << endl;

int tablouFinal[finalSize];

cout << endl;

for (size\_t i = 0; i < finalSize; i++)

{

if (i < size1)

{

tablouFinal[i] = tablou1[i];

}

else

{

tablouFinal[i] = tablou2[i - size1];

}

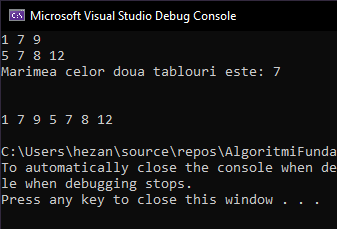
}

cout << endl;

printArray(tablouFinal, finalSize);

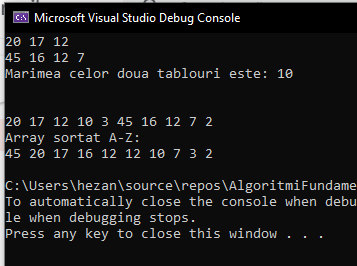
return 0;

}



# Exercitiul 2

Tablourile unidimensionale A și B au elementele: A=(20,17,12,10,3), iar B=(45,16,12,7,2) . În urma interclasării lor în ordine descrescătoare se obține tabloul cu elementele: (45,20,17,16,12,12,10,7,3,2)



void swap(int\* xp, int\* yp)

{

int temp = \*xp;

\*xp = \*yp;

\*yp = temp;

}

void sortZA(int arr[], int n)

{

int i, j, min\_idx;

for (size\_t i = 0; i < n - 1; i++)

{

min\_idx = i;

for (size\_t j = i + 1; j < n; j++)

{

if (arr[j] > arr[min\_idx])

{

min\_idx = j;

}

}

swap(&arr[min\_idx], &arr[i]);

}

}

int main()

{

int tablou1[] = { 20,17,12,10,3 }; // tablou A

int tablou2[] = { 45,16,12,7,2 }; // tablou B

printArray(tablou1,3);

printArray(tablou2,4);

const int size1 = sizeof(tablou1) / sizeof(tablou1[0]); // marime tablou A

const int size2 = sizeof(tablou2) / sizeof(tablou2[0]); // marime tablou B

const int finalSize = size1 + size2;

cout << "Marimea celor doua tablouri este: " << finalSize << endl;

int tablouFinal[finalSize];

cout << endl;

for (size\_t i = 0; i < finalSize; i++)

{

if (i < size1)

{

tablouFinal[i] = tablou1[i];

}

else

{

tablouFinal[i] = tablou2[i - size1];

}

}

cout << endl;

printArray(tablouFinal, finalSize);

sortZA(tablouFinal,finalSize);

cout << "Array sortat A-Z: " << endl;

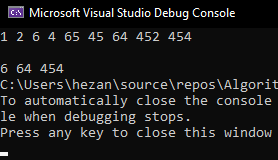
printArray(tablouFinal, finalSize);

return 0;

}

# Exercitiul 3

Thomas iubește numerele pare. Era plictisit și a decis să tipărească toate numerele pare la indicii pari ai unui tablou unidimensional. Vrea ajutorul tău pentru a finaliza această sarcină.



void printArray(int arr[], int size)

{

int i;

for (i = 0; i < size; i++)

cout << arr[i] << " ";

cout << endl;

}

int main()

{

int tablou1[] = { 1,2,6,4,65,45,64,452,454 };

int size1 = sizeof(tablou1) / sizeof(tablou1[0]);

printArray(tablou1,size1);

cout << endl;

for (size\_t i = 0; i < size1; i++)

{

if (i % 2 == 0)

{

if (tablou1[i] % 2 == 0)

{

cout << tablou1[i] << " ";

}

}

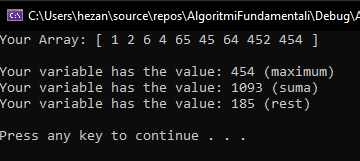
}

return 0;

}

# Exercitiul 4

Vi se dă un tablou de numere întregi și trebuie să găsiți suma elementelor din tablou și să găsiți restul atunci când suma este împărțită la cel mai mare număr din tablou.



void printArray(int arr[], int size)

{

int i;

cout << "Your Array: [ ";

for (i = 0; i < size; i++)

cout << arr[i] << " ";

cout << "]";

cout << endl;

}

int main()

{

int tablou1[] = { 1,2,6,4,65,45,64,452,454 };

int size1 = sizeof(tablou1) / sizeof(tablou1[0]);

printArray(tablou1,size1);

cout << endl;

int suma{ 0 }, rest{ 0 }, max{0};

max = tablou1[0];

for (size\_t i = 0; i < size1; i++)

{

if (tablou1[i] > max)

{

max = tablou1[i];

}

suma += tablou1[i];

}

printVar(max, "maximum");

printVar(suma, "suma");

rest = suma % max;

printVar(rest, "rest");

cout << endl;

system("pause");

return 0;

}

# Exercitiul 5

Se citeşte un şir X de numere naturale cu n elemente. Scrieţi un program care determină şirul Y de numere prime distincte, care figurează la puterea întâi în cel puţin o descompunere ȋn factori primi a unui număr din șirul X. Dacă niciun element al şirului X nu are un factor prim la puterea întâi, atunci se va tipări mesajul Sirul Y este vid.

Se vor scrie subprograme pentru:

* citirea unui şir de numere naturale
* tipărirea unui şir
* generarea tuturor numerelor prime mai mici sau egale decât un număr dat SAU verificarea dacă un număr este prim (ȋn funcție de modalitatea de rezolvare aleasă)
* verificarea dacă un număr figurează la puterea întâi în descompunerea unui număr dat
* construirea șirului Y

void printVar(int yourVariable, string yourText)

{

cout << "Your variable has the value: "<< yourVariable << " (" << yourText << ") ";

cout << endl;

}

void ReadArray(int someArray[], int size)

{

for (int i = 0; i < size;i++)

{

cout << endl << "Array[" << i << "]=";

cin >> someArray[i];

}

}

void printArray(int arr[], int size)

{

int i;

cout << "Your Array: [ ";

for (i = 0; i < size; i++)

cout << arr[i] << " ";

cout << "]";

cout << endl;

}

bool isPrim(int yourVar)

{

bool isPrim = true;

if (yourVar == 2)

{

isPrim = true;

}

for (size\_t i = 3; i < yourVar / 2; i = i + 2)

{

if (yourVar % i == 0 && isPrim == true)

{

isPrim = false;

}

}

return isPrim;

}

int main()

{

int arr[20];

int arrY[20];

int length{0};

int n{ 0 }, d{ 0 }, p{ 0 }, copyVal{ 0 }, count{ 0 };

cout << "Dimensiune array"<< endl;

addValueToVar(length);

cout << endl;

cout << "Elemente array: " << endl;

cout << "Array x: " << endl;

ReadArray(arr,length);

cout << endl;

printArray(arr, length);

d = 2;

// descompunere in factori primi si salvare in arrY

for (size\_t i = 0; i < length; i++)

{

copyVal = arr[i];

while (copyVal > 1)

{

p = 0;

while (copyVal % d == 0)

{

p++;

copyVal = copyVal / d;

}

if (p == 1 && isPrim(d) == true)

{

cout << d << "^" << p << " ";

arrY[count++] = d;

}

d = d + 1;

}

}

cout << "Array y: " << endl;

printArray(arrY, count);

system("pause");

return 0;

}

