|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ლექტორის აკად. ხარისხი:** | | | | ასოცირებული პროფესორი | | | | | **სახელი:** | | | | ლელა | | | **გვარი:** | | | გაჩეჩილაძე | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | |
| **საფეხური:** | | ბაკალავრიატი | | | | | **პროგრამა:** | | | | | კომპიუტერული მეცნიერება | | | | | | | | | | | | | | |
| **აკადემიური წელი:** | | | 2023-2024 | | | | **სწავლების ენა:** | | | | | ქართული | | | | | | **თარიღი:** | | | ოთხშაბათი, 24 იანვარი, 2024 | | | | | |
| **სემესტრი:** | შემოდგომის | | | | | **სასწავლო კურსის დასახელება:** | | | | | დაპროგრამების საფუძვლები | | | | | | | | | **ხანგრძლივობა:** | | | 2 სთ | **ჯგუფი:** | | 3 | |
| **კურსის კოდი:** | | CS104 | | | **ტიპი:** | | | დასკვნითი | | | | | | **ვარიანტი:** | 3 | | | | | | | **მაქსიმალური ქულა:** | | | 40 | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | |
| **სტუდენტის საიდენტ. ბარათის ნომერი:** | | | | | | | |  | | **სახელი, გვარი და ხელმოწერა:** | | | | | | |  | | | | | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **სტუდენტს უფლება აქვს გამოცდის მსვლელობისას თან იქონიოს (განისაზღვრება ლექტორის მიერ წინასწარ )[[1]](#footnote-1):** | | | | | |
| Choose an item. |  | Choose an item. |  | Choose an item. | **სხვა:** | |

1. (**5 ქულა**) მოცემულია სამი *a*, *b* და *c* მთელი ტიპის ცვლადი. გამოთვალეთ მათ შორის მინიმალური მნიშვნელობის ცვლადი (გამოიყენეთ მომხმარებლის მიერ შექმნილი **inline** სპეცი­ფიკაციის ფუნქცია და გამოიძახეთ ეს უკანასკნელი ჯერ უარგუმენტებოდ და შემდეგი ორი არგუმენტით).

#include <bits/stdc++.h>

using namespace std;

inline int max\_num(int a,int b, int c){

return a>b? a>c? a:c:b>c? b:c;

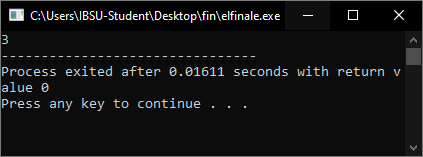
}

int main(int argc, char\*\* argv) {

cout << max\_num(1,2,3);

return 0;

}



1. (**5 ქულა**) შეადგინეთ პროგრამა, რომელიც **სამომხმარებლო ფუნციის** გამოყენებით ცალ-ცალკე განსაზღვრავს თქვენთვის საინტერესო ნებისმიერი ორი სიმბოლოს რაოდენობას თქვენ მიერ **დინამიურად ინიციალებულ** **სტრიქონში** და შესაბამისი ფუნქციის გამოყენებით წარმოგვიდგენს სტრიქონის **ზომას ბაიტებში.**

#include <bits/stdc++.h>

using namespace std;

int size\_of\_str(string str){

return str.size();

}

int find\_chars(string str,char ch){

int ans[2];

int len = size\_of\_str(str), n=0;

for(int i=0;i<len;i++){

if (str[i]==ch) n++;

}

return n;

}

int main(int argc, char\*\* argv) {

string text;

char ch1,ch2;

cin>>text;

cin>>ch1>>ch2;

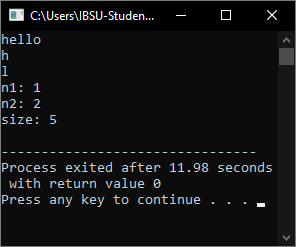
int n1=find\_chars(text,ch1);

int n2=find\_chars(text,ch2);

cout<<"n1: "<<n1<<"\nn2: "<<n2<<"\nsize: "<< size\_of\_str(text)<<endl;

return 0;

}



1. (**6 ქულა**) შეადგინეთ პროგრამა, რომელიც **მომხმარებლის მიერ შექმნილი ფუნქციების** გამოყე­ნებით int M[15] მასივის ელემენტებს მიანიჭებს ნებისმიერ მთელრიცხვა მნიშვნელობებს, დაბეჭდავს მას **ჰისტროგრამასთან** ერთად და **ბინარული ძებნის ალგორითმის** მიხედვით მოძებნის საჭირო ელემენტს მოცემულ მასივში (კონსოლზე გამოიტანეთ ამოცანის შესრულების ყველა ეტაპის შედეგები).

#include <bits/stdc++.h>

using namespace std;

void fill\_arr(int \*arr, int n, int max=10, int min=0){

srand(time(0));

cout << "Current array: ";

for (int i = 0; i < n; i++) {

arr[i]=min+rand()%max+1;

cout<<setw((int)(log10(max)+2)) << arr[i]<<",";

}

printf("\n");

}

void histogram(int \*arr, int n, char symbol='\*'){

cout<<"\nHistogram:\n";

for (int i=0;i<n;i++){

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

for(int j=0;j<arr[i];j++){

cout<<symbol;

}

cout << endl;

}

}

void insertion\_sort(int \*arr, int n){

int j;

for(int i=1;i<n;i++){

int tmp=arr[i];

for( j=i-1;j>=0&&arr[j]>tmp;j--) arr[j+1]=arr[j];

arr[j+1]=tmp;

}

}

int binary\_search(int \*arr, int n, int tar){

int low=0;

int high=n-1;

while (low<=high) {

int mid = (high+low)/2;

int val = arr[mid];

if (val < tar)low=mid+1;

else if (val > tar) high=mid+1;

else return mid;

}

return -1;

}

int main (int argc, char \*argv[]) {

int n=15;

int m[n];

fill\_arr(m,n);

histogram(m,n);

insertion\_sort(m,n);

cout << "\nSorted array: ";

for (int i=0; i<n;i++) cout<<setw((int)log10(m[n-1])+2)<<m[i];

cout << "\nWhat are you looking for? ";

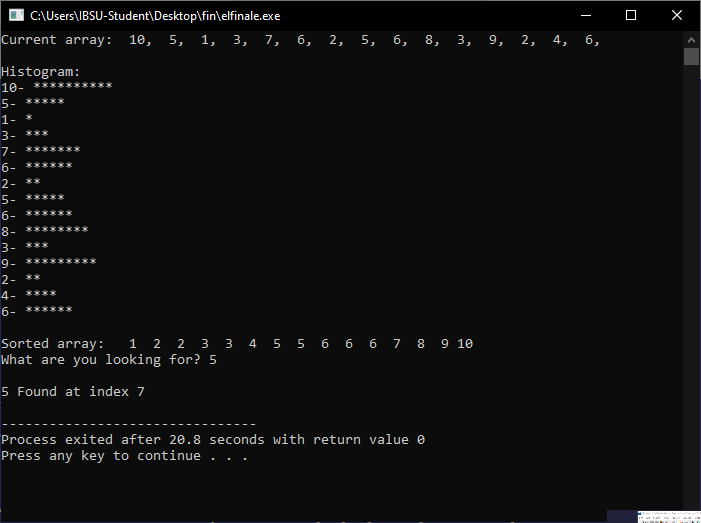
int tar;

cin >> tar;

cout << "\n"<<tar<<" Found at index "<<binary\_search(m,n,tar)<<endl;

return 0;

}



1. (**6 ქულა**) **სამომხმარებლო ფუნქციებისა და ფაილებში მონაცემების ჩაწე­რა/წაკითხვის საშუალებათა გამოყენებით** შეადგინეთ შემდეგი ამოცანის გადაწყვეტის პროგრამა: მოცემულია 4x4 განზო­მილების მთელრიცხვა N მატრიცა. მატრიცის ელემენტებს მიანი­ჭეთ ნებისმიერი მთელრიცხვა მნიშვ­ნელობები, განსაზღვრეთ მასში მინიმალური მნიშვნე­ლობის ელემენტი და ადგილები შეუცვალეთ ამ ელემენტის შემცველ სვეტისა და სტრიქონის ელემენტებს. ამოცანის საწყი­სი მონაცემები შეიტანეთ **myfile1.txt** **მიმდევრობითი მიმართვის ფაილით,** ხოლო საწყისი, სახეშეცვლილი მატრიცები (მატრიცების სახით) და მიღე­ბული შე­დე­გი გამოიტანეთ **myfile2.txt** **მიმდევ­რობითი მიმართვის ფაილში.41**

#include <bits/stdc++.h>

using namespace std;

const int ROW=4;

const int COL=4;

void fill\_mat(int mat[][COL]){

ifstream fin("myFile1.txt");

for (int i=0;i<ROW;i++){

for (int j=0;j<COL;j++){

fin>>mat[i][j];

}

}

fin.close();

}

int\* min\_index(int mat[][COL]){

int m=mat[0][0],mi=0,mj=0;

for (int i=0;i<ROW;i++){

for (int j=0;j<COL;j++){

if (m<mat[i][j]) {

m=mat[i][j];

mi=i;

mj=j;

}

}

}

int ans[2]={mi,mj};

return ans;

}

void swap\_mat(int mat[][COL],int mi,int mj){

for (int i=0;i<ROW;i++){

int tmp=mat[i][mj];

mat[i][mj]=mat[mj][i];

mat[mj][i]=tmp;

}

}

void save\_matrix(int mat[][COL]){

ofstream fout("myFile2.txt");

for (int i=0;i<ROW;i++){

for (int j=0;j<COL;j++){

fout<<setw(4)<<mat[i][j];

}

fout<<"\n";

}

fout.close();

}

int main (int argc, char \*argv[]) {

int mat[ROW][COL];

fill\_mat(mat);

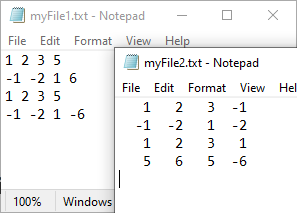
int \*ans=min\_index(mat);

swap\_mat(mat,ans[0],ans[1]);

save\_matrix(mat);

return 0;

}



1. (**6 ქულა**) მოცემულია 20 ელემენტისგან შემდგარი მთელრიცხვა int D[20] მასივი. **სამომხმარებლო ფუნქციების** გამოყენებით შეადგინეთ პროგ­რამა, რომელიც მასივის ელემენტებს [-12;18] რიცხვითი დიაპაზონიდან მიანიჭებს მთელი ტიპის შემთხვევით მნიშვნელობებს და *m* რიცხვზე დამოკი­დე­ბუ­ლებით გამოთვლის მასივის დადებითი მნიშვნელობის ელემენტების საშუალო არითმეტიკულს იმ შემ­თხვე­ვა­ში, თუ *m<*0, წინააღმდეგ შემთხვევაში – გამოთვლის მასივის უარყო­ფითი მნიშვნელობის ელემენტების საშუალო არითმეტიკულს. პროგრამის ყველა ფუნქციაში (სადაც მოცემული მა­სი­ვია გამოყენებული) ერთ-ერთი პარამეტრის სახით განიხი­ლეთ **მიმთითებელი მასივზე**, ხოლო ჯამის ფუნქციაში – **მიმთითებელი ფუნქციაზე**.

#include <bits/stdc++.h>

using namespace std;

void fill\_array(int \*arr, int n,int min=0,int max=10){

srand(time(0));

cout<<"Current array:";

for (int i = 0; i < n; i++) {

arr[i]=min+rand()%(max\*2+1);

cout<<setw((int)(log10(max)+2))<<arr[i];

}

printf("\n");

}

bool get\_m(int m){

return m>0;

}

double avg(int \*arr,int n, int m,bool (\*m\_get)(int)){

int s=0;

int g=0;

for (int i=0;i<n;i++){

if (m\_get(m) && arr[i]<=0) {s+=arr[i];g++;}

else {s+=arr[i];g++;}

}

return 1.0\*s/g;

}

int main (int argc, char \*argv[]) {

int n=20,m;

int d[n];

fill\_array(d,n,-12,18);

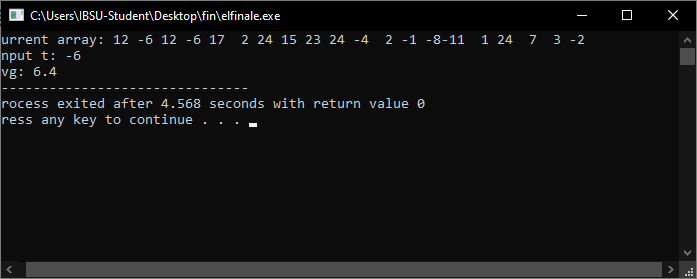
printf("Input t: ");

cin>>m;

cout<<"avg: "<<avg(d,n,m,get\_m);

return 0;

}



1. (**6 ქულა**) შეადგინეთ პროგრამა, რომელიც **სამომხმარებლო ფუნქციების** გამოყენებით **მეხსიერების დინამიკური დანაწილების** გათვა­ლისწი­ნე­ბით მოახდენს **ორგანზომილებიანი** მასივის ორგანიზებას: მასივის ელემენტებს მიანი­ჭებს მთელი ტიპის ნებისმიერ შემთხვევით მნიშვნელობებს, ცალ-ცალკე გამოთვლის მასივის დადებითი მნიშვნელობის ელემენ­ტების საშუალო გეომეტრიულს და უარყოფითი მნიშვნელობის ელემენ­ტე­ბის ნამრავლს. მასივ­თან ერთად კონ­სოლზე გამოიტანს ყველა შედეგს და პროგრამის დასას­რულს მოახდენს **მეხსიე­რე­ბის დინამიკურ გამოთავისუფლებას.**

**#include <bits/stdc++.h>**

**using namespace std;**

**int rand\_num(int min,int max) {**

**return min+rand()%(max\*2+1);**

**}**

**bool is\_pos(int a){**

**return a>0;**

**}**

**int main (int argc, char \*argv[]) {**

**int row, col;**

**printf("Input rows: ");**

**cin>>row;**

**printf("Input collumns: ");**

**cin>>col;**

**srand(time(0));**

**int\*\* matrix = new int\*[row];**

**for(int i=0;i<row;i++){**

**matrix[i] = new int[col];**

**}**

**cout<<"Current matrix:\n";**

**for(int i=0;i<row;i++){**

**for(int j=0;j<col;j++){**

**matrix[i][j]=rand\_num(-10,10);**

**cout<<setw(4)<<matrix[i][j];**

**}**

**printf("\n");**

**}**

**int g=0,p1=1,p2=1;**

**double avg\_geo=0;**

**for(int i=0;i<row;i++){**

**for(int j=0;j<col;j++){**

**if(is\_pos(matrix[i][j])){p1\*=matrix[i][j];g++;}**

**else p2\*=matrix[i][j];**

**}**

**}**

**avg\_geo=pow(p1,1/g);**

**cout<<"avg geo: "<<avg\_geo<<"\nproduct: "<<p2<<endl;**

**for(int i=0;i<row;i++){**

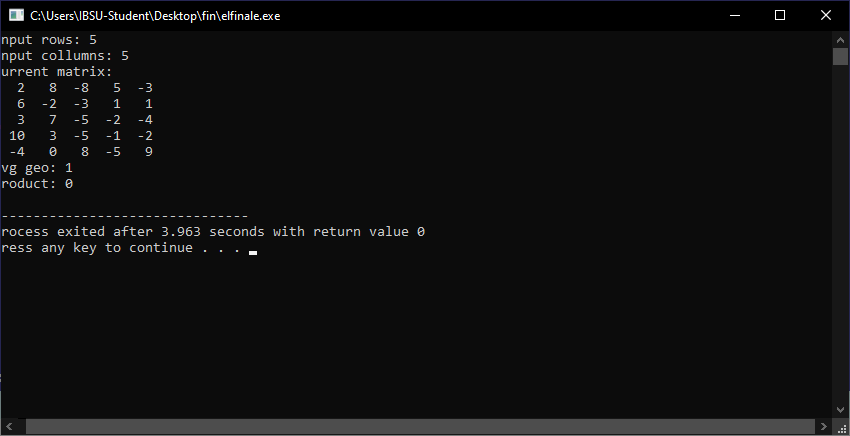
**delete[] matrix[i];**

**}**

**delete matrix;**

**return 0;**

**}**

****

1. (**6 ქულა**) მომხმარებლის მიერ შექმნილი **სტრუქტურის** გამოყენებით შეადგინეთ პროგრამა, რომელიც თანაბ­რად განაწილებული შემთხვევითი რიცხვების გენერირების გზით, D[6x6] მატრიცის ელემენტებს [20; 90] დიაპაზონიდან მიანიჭებს მთელი ტიპის კვაზი­შემთხვევით მნიშვ­ნელობებს. ცალ-ცალკე გამოთვლის მატრიცის კენტ და ლუწინდექსებიანი ელე­მენტების კვადრატების ჯამის მნიშვნელობებს და მიღებულ შედეგებს შეადარებს ერთმანეთს. ( სტრუქტურაში კვადრატების ჯამის გამოთვლის ფუნქცია-ელემენტების სახით განიხილეთ ფუნქციები, რომლებიც მიღებულ შედეგებს დააბრუნებენ main() ფუნქციაში).

**#include <bits/stdc++.h>**

**using namespace std;**

**struct Mat {**

**int arr[4][4];**

**int ans[2];**

**int\* add(){**

**int se=0;**

**int so=0;**

**for(int i=0;i<4;i++){**

**for (int j=0;j<4;j++){**

**if(i%2 && j%2) so+=arr[i][j]\*arr[i][j];**

**else se+=arr[i][j]\*arr[i][j];**

**}**

**}**

**ans[0]=se;ans[1]=so;**

**return ans;**

**}**

**void fill(int min,int max){**

**for (int i = 0; i < 4; i++) {**

**for (int j = 0; j < 4; j++) {**

**arr[i][j]=min+rand()%(max+1);**

**}**

**}**

**}**

**};**

**template <typename T>**

**T compare(T a,T b){**

**return a>b? 0:1;**

**}**

**int main (int argc, char \*argv[]) {**

**srand(time(0));**

**Mat matx;**

**matx.fill(20,90);**

**for (int i = 0; i < 4; i++) {**

**for (int j = 0; j < 4; j++) {**

**cout<<setw(4)<<matx.arr[i][j];**

**}**

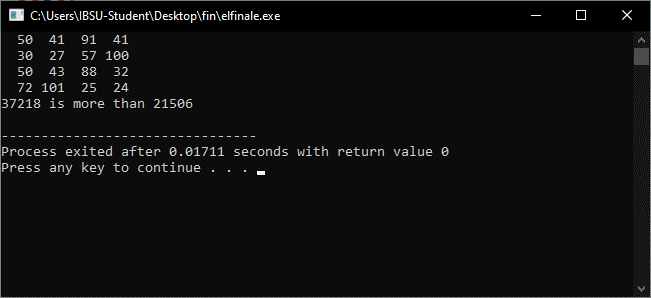
**cout<<endl;**

**}**

**cout<<matx.ans[compare(matx.add()[0],matx.add()[1])]<<" is more than "<<matx.ans[compare(matx.add()[1],matx.add()[0])]<<endl;**

**return 0;**

**}**



1. ივსება ლექტორის მიერ! ცვლილება უნდა გაკეთდეს მხოლოდ **ელექტრონულად**, დოკუმენტის ამობეჭდვამდე. ხელით მინაწერი მხედველობაში არ მიიღება! [↑](#footnote-ref-1)