**13-14-Laboratoriya mashg’uloti**

**Mavzu: Massiv tushunchasi. Mutaxassislik masalalarini massivlar yordamida algoritmlash va dasturlash. Bir o’lchovli massivlar**

**Kerakli texnik vositalar:** Pentium - 4 shaxsiy kompyuteri.

**Kerakli dasturiy vositalar:** C++ muhiti.

**Mashulotning maqsadi:** Bir va ikki oʻlchovli massivlarni dasturlash, ulardan mutaxassislik masalarini yechishda foydalanish ko’nikmasini hosil qilish.

**Bir o’lchovli massivlarni dasturlash bo’yicha topshiriqlar variantlari:**

1. a)  va vektorlarning skalyar koʻpaytmasini aniqlang.

b)  matritsa elementlari orasidan eng kichigini aniqlang.

a)  va  vektorlar yigʻindisidan tashkil topgan vektorning eng katta koordinatasini aniqlang.

b)  matritsaning eng katta va eng kichik elementlari yigʻindisini toping.

1. a)  vektordan  vektorini hosil qiling va uni  vektor bilan hosil qilgan burchak kosinusini hisoblang.

b) matritsani  vektorga koʻpaytirishdan hosil boʻlgan vektorni aniqlang.

1. a)  vektor berilgan boʻlsa, vektor uzunligini hisoblang.

b)  matritsaning diagonal elementlaridan tuzilgan vektor uzunligini aniqlang.

1. a)  vektor koordinatalari orasidan eng kichigining indeksi (tartib nomeri)ni aniqlang.

b)  va  matritsalarni koʻpaytmasini hisoblang.

1. a)  matritsaning musbat elementlari oʻrniga 1 sonini, manfiy elementlari oʻrniga esa –1 sonini joylashtiring.

b)  matritsaning manfiy elementlari sonini aniqlang.

1. a)  vektor koordinatalari orasidan eng kattasining indeksi (tartib nomeri)ni aniqlang.

b)  matritsaning 1 dan kichik elementlari yigʻndisini aniqlang.

1. a)  va  vektorlar berilgan boʻlsa,  koʻpaytmani hisoblang.

b)  matritsaning eng katta va eng kichik elementlari koʻpaytmasini aniqlang.

1. a)  vektor elementlarini kamayish tartibida joylashtiring.

b)  matritsaning absolyut qiymati 3 dan kichik boʻlgan elementlari soni va ularning koʻpaytmasini aniqlang.

1. a)  vektorning musbat elementlarini oʻrta arifmetigini hisoblang.

b)  matritsaning 1 dan kichik elementlari oʻrniga 0 sonini, 1 dan katta elementlari oʻrniga esa 1 sonini joylashtiring.

**Ikki o’lchovli massivlarni dasturlash bo’yicha topshiriqlar variant:**

**1.** A(3,4) matritsaning musbat elementlari soni topilsin.

**2.** B(4,5) matritsaning har bir satridagi manfiy elementlari soni topilsin.

**3.**A(3,4) matritsaning har bir ustunidagi juft sonlar nechtaligini aniqloqchi dastur tuzing.

**4.**B(4,5) matritsaning juft manfiy elementarinng nechtaligini aniqlang.

**5.**A(3,4) massivning eng katta elementini hisoblash dasturini tuzing.

**6.** A(3,4) massivning eng katta elementi tartib nomerini xisoblash dasturini tuzing.

**7.** A(3,4) massivning eng kichik elementini xisoblash dasturini tuzing.

**8.** B(4,5) matritsaning har bir ustunidagi elementlari yig‘indisi topilsin.

**9.** B(4,5) matritsaning har bir ustunidagi elementlari yig‘indisi topilsin.

**10.** B(4,5) matritsaning har bir ustunidagi elementlari ko‘paytmasi topilsin.

**11.** B(4,5) matritsaning har bir satridagi elementlari ko‘paytmasi topilsin

**12.** S(2,4) massivning diagonal elementlari ko‘paytmasi topilsin.

**13.**S(4,4) massivda eng kam musbat elementlarni o‘z ichiga oluvchi ustun nomerini aniqlang.

**14.** S(4,4) massivda eng ko‘p juft sonlarni o‘z ichiga oluvchi satrni aniqlang.

**15.** A(5,5) massivda 5 ga karrali eleentlar sonini aniqlang.

**16.** B(4,5) massivda eng kichik juft soni toping.

**17.** B(4,5) massivda 3 ga karrali eng katta sonni toping.

**18.** B(5,5) massivning asosiy diagonilaida yotgan elementlarining sonini va yig‘indisini toping

**19.**B(4,4) massivning musbat elementlarini 0 ga almashtiring va yangi xosil bo‘lgan matritsani yozing.

**20.**B(4,5) massivning xar bir qatoridagi maksimal va minal elementlarini toping va ularning o‘rnini almashtiring

**Nazariy qism**

Bir ulchovli massivlar. Massiv bu bir tipli nomerlangan ma'lumotlar jamlanmasidir. Massiv indeksli o’zgaruvchi tushunchasiga mos keladi. Massiv ta'riflanganda tipi, nomi va indekslar chegarasi ko’rsatiladi. Misol uchun long int a[5]; char w[200];double f[4][5][7]; char[7][200]. Massiv indekslar har doim 0 dan boshlanadi. C ++  tili standarti bo’yicha indekslar soni 31 tagacha bo’lishi mumkin, lekin amalda bir o’lchovli va ikki o’lchovli massivlar qo’llaniladi. Bir ulchovli massivlarga matematikada vektor tushunchasi mos keladi. Massivning int z[3] shakldagi ta'rifi, int tipiga tegishli z[0],z[1],z[2] elementlardan iborat massivni aniqlaydi. Massivlar ta'riflanganda initsializatsiya qilinishi, ya'ni boshlang’ich qiymatlarlari ko’rsatilishi mumkin. Misol uchun: float C[]={1,-1,2,10,-12.5};

Bu misolda massiv chegarasi avtomatik aniqlanadi. Agar massiv initsializatsiya qilinganda elementlar chegarasi ko’rsatilgan bo’lsa , ruyhatdagi  elementlar soni bu chegaradan kam bo’lishi mumkin, lekin ortiq bo’lishi mumkin emas. Misol uchun int A[5]={2,-2}. Bu holda a[0] va a[1] qiymatlari aniqlangan bo’lib, mos holda 2 va –2 ga teng.

Massivda musbat elemenlar soni va summasini hisoblash.

• # include <iostream.h>;

• # include <conio.h>;

• Main() {

• Int x[]={-1;2;5;-4;8;9};

• Clrscr();

• For (int s=0,int k=0, int I=0; I<6; I++) {

• If (x[I]<=0) continue;

• k++;s++;

• };

• Cout<<(“%d”,k);

• Cout<<(“%d”,k);

• getch();

• }

• Cout<<(“%d”,k);

• Cout<<(“%d”,k);

• getch();

• };

• };

Massivning eng katta, eng kichik elementi va o’rta qiymatini aniqlash:

• #include <iostream.h>

• Void main()

• {

• Int I,j,n;

• Float a,b,d,x[100];

• While(1)

• {

• Cout<<(“\n n=”); Cin>>(“%i”,&n);

• If ( n>0 && n < = 100 ) break;

• Cout<<(“\n Hato 0<n<101 bulishi kerak”);

• }

• Cout<<(“\n elementlar kiymatlarini kiriting:\n”);

• For (i=0;i<n;i++)

• { Cout<<(“x[%i]=”,i);Cin>>(“%f”,&x[i]);}

• max=x[0];min=x[0];

• For (s=0,i=0;i<n;i++)

• { s++;

• If (max<x[i]) max=x[i];

• If (min>x[i]) min=x[i];

• };

• s/=n;

• Cout<<(“\n max=%f”,max);

• Cout<<(“\n min=%f”,min);

• Cout<<(“\n urta kiymat=%f”,s);

• }

Massiv - bir xil turdagi ma'lumotlar to'plamidir. Massivg quyidagi ko`rinishda yoziladi: ***o'zgaruvchi\_turi massiv\_nomi [massiv\_uzunligi].*** O'zgaruvchining turidan keyin massiv nomi, so'ngra kvadrat qavslarda uning o'lchamlari ko`rsariladi. Masalan, 4 ta sondan iborat massivni aniqlaymiz: **int mas [4].** Ushbu massiv to'rtta elementga ega, ammo ularning qiymatlari aniqlanmagan.Bu elementlarga o`rta qavslar orqali boshlang'ich qiymatlarni berishimiz mumkin: ***int mas [4] = {1,2,3,4};***

Agar massiv o`lchamiga nisbatan qiymatlar kamroq bo'lsa, unda qiymatlar massivning dastlabki elementlariga beriladi. Agar massiv o`lchamiga nisbatan qiymatlar ko`proq bo'lsa, unda kompilyatsiya paytida xato yuzaga keladi: ***int mas [4] = {1, 2, 3, 4, 5, 6};***. Bu erda massiv 4 o'lchamga ega, ammo unga 6 qiymat berilgan. Agar massivning o'lchami aniq ko'rsatilmagan bo'lsa, unda o`lcham berilgan qiymatlar soniga qarab aniqlanadi: *int mas [] = {1, 2, 3, 4, 5, 6};.* Bu holda, massivda 6 element mavjud bo`lib, massiv o`lchami ham 6 ga teng.

Belgilar massivlarini initsializatsiyalash (qiymat berish) o'ziga xos xususiyatlarga ega. Ushbu belgili massivda biz boshlang'ich qimatni belgilar orqali ham, satr ko`rinishida ham berishimiz mumkin:

***char s1 [] = {'h', 'e', 'l', 'l', 'o'};***

***char s2 [] = "dunyo";***

Bundan tashqari, ikkinchi holatda, s2 massivida 5 ta emas, balki 6 ta elementga ega bo`ladi. Chunki belgili massivga satr orqali qiymat berganda massiv oxiriga avtomatik ravishda '\ 0' nol belgisi qo'shiladi. Shu bilan birga, bitta massiv elementlarini boshqa massivga o`lashtirishga ruxsat berilmaydi:

***int nums1 [] = {1,2,3,4,5};***

***int nums2 [] = nums1; // xato***

***nums2 = nums1; // xato***

Massivni aniqlagandan so'ng, indeks orqali uning elementlariga murojaat qilishimiz mumkin. Indekslar noldan boshlanadi, shuning uchun birinchi elementga kirish uchun 0 indeksidan foydalanish kerak. Elementga indeks orqali murojaat qilib, uning qiymatini olishimiz yoki o'zgartirishimiz mumkin:

**1-misol.**

***#include <iostream>***

using namespace std;

***int main()***

***{***

int numbers[4] = {1,2,3,4};

int first\_number = numbers[0];

cout << first\_number<<endl; // 1

***numbers[0] = 34;***

cout << numbers[0] << endl; // 34

return 0;

***}***

Sikldan foydalanib, massivning elementlariga indekslar orqali murojaat qilish mumkin:

**2-misol.**

***#include <iostream>***

using namespace std;

***int main()***

***{***

***int numbers[4] = {1,2,3,4};***

int size = sizeof(numbers)/sizeof(numbers[0]);

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

cout << numbers[i] << endl;

return 0;}

Sikl yordamida massivni bo'ylab yurish uchun avval massivning uzunligini topish kerak. Uzunlikni topish uchun sizeof operatoridan foydalaniladi. Aslida, massiv uzunligi uning elementlarining umumiy uzunligiga teng. Barcha elementlar bir xil turda va xotirada bir xil hajmni egallaydi. Shunday qilib, sizeof (numbers) ifodasidan foydalanib, butun massiv uzunligini baytlarda topamiz va sizeof (numbers [0]) yordamida bitta element uzunligini baytlarda topamiz.Birinchi qiymatni ikkinchisiga bo`lib, massivdagi elementlar sonini aniqlash mumkin. Soo`ng for operatoridan fodalanib, i hisoblagich qiymati massiv uzunligiga teng bo'lgunga qadar barcha elementlarga murogaat qilamiz.

**3-misol.** Haqiqiy ***a(n), n≤15*** massivning barcha musbat elementlarining o‘rta arifmetigini hisoblang.

Belgilashlarni kiritamiz. ***i -*** massiv elementlari indesklarining qiymatlarini aniqlovchi o‘zgaruvchi, ***i=1,2,..k (k*** - massiv elementlari soni, ***k******15), S*** va ***kn*** mos ravishda musbat elementlarning o‘rta arifmatik qiymati va ularning soni. ***a[i]>0*** shart bajarilsa ***S=S+a[i], kn=kn+1*** yig‘indi hisoblanishi kerak. Ushbu yig‘indini hisoblash tugagandan keyin ***S=S/kn*** hisoblanadi, ya’ni musbat elementlar yig‘indisi musbat elementlar soniga bo‘linadi. Agar ***kn=0*** bo‘lsa, massivning musbat elementlari mavjud emas.

***#include <iostream>***

using namespace std;

***int main()***

***{***

***int i,k,kn;***

***float s;***

***cout<<"k ni kiriting \n"; cin>>k;***

***float a[k]; s=0; kn=0;***

cout<<"a massivni kiriting \n";

***for (i=1;i<=k; i++) cin>>a[i];***

***for (i=1;i<=k; i++)***

***if (a[i]>0)***

***{s=s+a[i]; kn=kn+1; }***

***if (kn==0){***

***cout<<"Musbat elementlari yo`q \n";***

***goto t;}***

***else***

***{s=s/kn;***

***cout<<"s="<<s;***

***}***

***t:;}***

**4-misol.** Bir o‘lchovli haqiqiy ***B(N), n=20*** massivning eng katta elementini va uning tartib raqamini toping.

Belgilashlar: ***bmax*** va ***imax –B(N)*** massivning eng katta elementi va uning tartib raqami; ***k*** – massiv elementlari soni bo‘lsin.

Dastlab ***b*** massivning barcha elementlari qiymatini kiritamiz. Massiv birinchi elementini eng katta element deb faraz qilamiz: ***bmax=b1, imax=1***. ***i*** sikl parametri ***2*** dan ***k*** gacha o‘zgaradi. Har gal ***bi >bmax*** shart tekshiriladi, agar bu shart bajarilsa ***bmax=bi,*** va ***imax=i*** almashtirishlar bajariladi.

***#include <iostream>***

using namespace std;

***int main()***

***{***

***// const int n=5;***

***int i,k,imax;***

***float bmax;***

***//float b[n];***

***cout<<"k ni kiriting"<<endl;***

***cin>>k;***

***float b[k];***

***cout<<"b massivni kiriting \n";***

***for (i=1;i<=k;i++)cin>>b[i];***

***{bmax=b[1]; imax=1;}***

***for (i=1;i<=k;i++)***

***{***

***if (b[i]>bmax)***

***{bmax=b[i]; imax=i;}***

***}***

***cout<<"bmax="<<bmax<<"\t imax="<<imax;***

***}***

**Ikki o‘lchovli massivlar**

C ++ da bir o'lchovli massivlardan tashqari, ko'p o'lchovli massivlar ham mavjud. O'z navbatida, bunday massivlarning elementlari massiv bo'lib, unda elementlar ham massiv bo'lishi mumkin. Masalan, ikki o'lchovli massivni aniqlaymiz:

***int numbers [3] [2];***

Bunday massiv uchta elementdan iborat bo'lib, har bir element ikkita elementdan iborat massivdir. Bunday Massivga quyidagicha qiymat beramiz:

***int numbers [3] [2] = {{1, 2}, {4, 5}, {7, 8}};***

Ichki qavslar har bir qism massiv uchun elementlarni belgilaydi. Bunday massivni jadval shaklida ham ifodalsh mumkin:

1 2

4 5

7 8

Qiymat berishda ichki qavslarni yozmaslik mumkin:

***int numbers [3] [2] = {1, 2, 4, 5, 7, 8};***

Ikki o`lchovli massiv elementlariga murojaat qilish uchun ikkita indeks kerak:

***int numbers [3] [2] = {{1, 2}, {3, 4}, {5, 6}};***

***cout << numbers [1] [0] << endl; // 3***

numbers [1] [0] = 12; // elementni o'zgartirish

cout << numbers[1] [0] << endl; // 12

1-misol.

***#include <iostream>***

***using namespase std;***

***int main()***

const int rows = 3, columns = 2;

***int numbers[rows][columns] = { {1, 2}, {3, 4}, {5, 6} };***

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

{

for(int j=0; j < columns; j++)

{

***cout << numbers[i] [j] << "\t";***

}

cout <<endl;

}

return 0;

**2-misol.** Ikki o‘lchovli haqiqiy ***B(N,M)*** massiv musbat elementlari yig‘indisini hisoblashga dastur tuzing ***(N50, M***≤***30).***

Belgilashlar: ***S*** –massiv elementlari yig‘indisi; ***k,l*** – mos ravishda massivning satr va ustunlar soni; ***n,m*** – massiv satr va ustunlarining maksimal soni, ***i*** –satr raqami***, j –*** ustun raqami bo’lsin.

Ikki o‘lchovli massiv ishtirok etgan masalalarni yechishda ichma-ich joylashgan sikllarni tashkil qilish maqsadga muvofiqdir. Satr va ustunlarni tashkil qilish uchun ***i*** va ***j*** o‘zgaruvchilarni kiritamiz. ***I*** tashqi sikl parametri, ***j*** esa ichki sikl parametri bo‘lishi lozim. ***I*** va ***j*** ning har bir qiymatida ***bij>0*** shart tekshiriladi, agar shart bajarilsa ***S=S+bij*** yig’indi hisoblanadi. Agar ***S=0*** bo‘lsa massivning musbat elementlari mavjud emas.

***#include <iostream>***

using namespace std;

***int main()***

***{ const int n=2, m=3;***

***int i,j,k,l;***

***float s;***

***float b[n][m];***

***s=0;***

***cout<< "b massivni kiriting \n";***

***for (i=0;i<n;i++)***

***for (j=0;j<m;j++)***

{

cin>>b[i][j];

if (b[i][j]>0) s=s+b[i][j];

***}***

***cout<<"s= \t"<<s;***

***if (s=0) cout<<"b massivning musbat elementlari yo`q";***

return 0;}

**3-misol.** Haqiqiy ikki o‘lchovli massivning eng katta elementini toping.

Belgilashlar: ***b*** – massiv nomi; , ***i=20, j=15*** – mos ravishda massivning satrlar va ustunlarining maksimal soni; ***k*** va ***l*** esa satr va ustunlar soni.

Dastlab ***b*** massivning barcha elementlari qiymatlarini kiritamiz. Massiv birinchi elementini eng katta element deb faraz qilamiz: ***bmax=b11***. Keyin esa ichma-ich siklning ichida ***bij >bmax*** shart tekshiriladi, agar bu shart bajarilsa ***bmax=bij*** almastirish bajariladi.

***#include <iostream>***

using namespace std;

***int main()***

***{***

***const int n=2,m=5;***

***int i,j,k,l;***

***float bmax;***

***float b[n][m];***

***cout<< "b massivni kiriting \n";***

***for (i=0;i<n;i++)***

***for (j=0;j<m;j++)***

cin>>b[i][j];

***bmax=b[1][1];***

***for (i=0;i<n;i++)***

***for (j=0;j<m;j++)***

***if (b[i][j]>bmax) bmax=b[i][j];***

***cout<<"bmax="<<bmax;***

***}***

**4-misol.** Ikki o‘lchovli haqiqiy ***B(N,M)*** massivning bosh diagonalining quyi qismida joylashgan elementlari yig‘indisini hisoblang ***(N***≤***50, M***≤***30)***.

Dasturning ko‘rinishi:

***#include <iostream>***

***using namespace std;***

***int main()***

***{***

***const int n=5, m=3;***

***int i,j,k,l;***

***float s;***

***float b[n][m];***

***s=0;***

***cout<< "b massivni kiriting \n";***

***for (i=0;i<n;i++)***

***for (j=0;j<m;j++)***

***{***

cin>>b[i][j];

if (i>j) s=s+b[i][j];

}

***cout<<"s="<<s;***

***}***

**Nazorat savollari**

1. Bir oʻlchovli massivlarni ta’riflang.
2. Massivlar Paskal tilida qanday ifodalanadi?
3. Bir oʻlchovli massiv musbat elementlarining oʻrta geometrigini hisoblashga dastur tuzing.
4. Bir oʻlchovli massiv elementlarining eng kichigini topishga dastur tuzing.
5. Ikki oʻlchovli massivlar, ularning dasturda ifodalanishini aytib bering.
6. Ikki oʻlchovli massiv manfiy elementlari sonini topish uchun dastur tuzing.
7. Ikki oʻlchovli massiv musbat elementlarining oʻrta arifmetigini topish uchun dastur tuzing.