**Funksiyalar bilan ishlash**

Assalomu alaykum bo`lajak dasturchi! Yangi mavzu video fayli bilan tanishib chiqing.  
Video faylni ko`rib bo`lgandan keyin ma`ruza matnini diqqat bilan o`qib chiqing.  
Dasturlash mobaynida bir xil ifodalarni, hisoblash jarayonlarini qayta – qayta hisoblashga to`g`ri keladi.  
Dasturlash tillarida, kompyuter hotirasini va dasturchining vaqtini tejash maqsadida, bunday takkorlanuvchi jarayonlarni dasturda ajratib yozib, unga asosiy daturdan, boshqa funksiyalardan murojaat qilish imkoniyatlari keltirilgan.  
Dasturning istalgan qismidan murojaat qilib, bir necha bor ishlatish mumkin bo`lgan operatorlar guruhiga funksiya deyiladi.  
C++ funksiyalar tili deyiladi. Chunki dasturda kamida bitta main funksiyasi bo'ladi. Asosiy dastur, asosiy funksiya deganda aynan manashu main funksiyasini tushunamiz.  
Asosiy programmadan (yoki chaqiruvchi funksiyadan) xech qanday parametr qabul qilib olmaydigan funksiyalarga, parametrsiz funksiyalar deyiladi.  
Parametrsiz funksiyaning o'zi ham 2 xil bo'lishi mumkin:  
1) Asosiy programmaga (yoki chaqiruvchi funksiyaga) natijani qaytaruvchi.  
2) void turidagi funksiya bo'lib, asosiy programmadan (yoki chaqiruvchi funksiyadan) xech qanday parametr qabul qilib olmaydi xam, asosiy programmaga xech qanday natija qaytarmaydi ham.  
Parametrsiz funksiyaga murojaat qilishda dastur tanasida funksiya nomi yoziladi. Dasturda funksiya nomi operatorlar kabi ishlatiladi. Parametrsiz funksiyada asosiy dasturning barcha global o`zgaruvchilaridan foydalanish mumkin.

Global o`zgaruvchilar  
Ham asosiy programmada, ham funksiyada ishlatish mumkin bo'lgan o`zgaruvchilar global o'zgaruvchilar deyiladi. Global o'zgaruvchilar asosiy programmada e`lon qilishi kerak.  
Lokal o`zgaruvchilar  
Faqat funksiyada ishlatish mumkin bo'lgan o'zgaruvchilarga local o'zgaruvchilar deyiladi. Ular funksiyada e`lon qilinadi. Funksiyada yana bir nechta ichki funksiyalardan foydalanish mumkin.  
Blok ichida e'lon qilingan o'zgaruvchilar, shu blok uchun lokal o'zgaruvchilar hisoblanadi. Bu o'zgaruvchilardan faqat blok ichida foydalanish mumkin.  
Parametrli funksiyalar  
Asosiy dasturdan (funksiyadan) chaqiriluvchi funksiyaga uzatilgan parametrlarni qabul qilib qayta ishlovchi funksiyalar parametrli funksiyalar deyiladi.  
Qiymat parametrlar  
Qiymat parametrlar – asosiy dasturdan funksiyaga uzatiladigan o'zgaruvchilar qiymatlarni qabul qilib oluvchi parametrlar. Funksiyaga murojaat qilinganida qiymat parametrlari uchun xotiradan joy ajratiladi. Funksiya tugaganida qiymat parametrlari uchun ajratilgan xotira bo'shatiladi.  
Ko'rsatkich parametrlar  
Ko'rsatkich parametrlar - asosiy dasturdan funksiyaga uzatiladigan o'zgaruvchilarning xotiradagi adresini qabul qilib oluvchi parametrlar.  
Ko'rsatkich parametrlari ustida bajarilgan har qanday o'zgarish, asosiy dasturdagi o'zgaruchilarning xotira adresida sodir bo'ladi. (Ya'ni asosiy dasturdagi o'zgaruvchi qiymati o'zgaradi)  
Eslatma: Qiymat parametrlari va ko'rsatkich parametrlar toifasi, asosiy dasturdagi qiymati uzatilayotgan o`zgaruvchilar toifasi bilan bir xil bo`lishi lozim.  
Funksiyadan chiqish  
Ixtiyoriy funksiyadan chiqish uchun return xizmatchi so'zi ishlatiladi.  
Misol: To'g'ri burchakli uchburchakning katetlari berilgan. (3, 4), (6, 8), (12, 5) bo'lgan xollar uchun uchburchak gipotenuzasini hisoblovchi programma tuzilsin.

1) Parametrli funksiya  
// Muallif: Qudrat Abdurahimov  
// Sana : 1 noyabr 2011 yil  
// Maqsad : Parametrli funksiyani o'rganish  
#include <iostream>  
#include <math.h>  
using namespace std;  
// funksiya prototipi  
float hisobla(float , float );  
int main()  
{  
float c;  
c = hisobla(3, 4);  
cout << c << endl;  
c = hisobla(6, 8);  
cout << c << endl;  
c = hisobla(12, 5);  
cout << c << endl;  
return 0;  
}  
float hisobla(float a, float b)  
{  
//lokal o'zgaruvchi  
float natija;  
natija = sqrtf(a\*a + b\*b);  
return natija;  
}  
2) void toifasidagi parametrli funksiya  
// Muallif: Qudrat Abdurahimov  
// Sana : 1 noyabr 2011 yil  
// Maqsad : void toifasidagi parametrli funksiyani o'rganish  
#include <iostream>  
#include <math.h>

using namespace std;  
// funksiya prototipi  
void hisobla(float , float );  
int main()  
{  
hisobla(3, 4);  
hisobla(6, 8);  
hisobla(12, 5);  
return 0;  
}  
void hisobla(float a, float b)  
{  
float c;  
c = sqrtf(a\*a + b\*b);  
cout << c << endl;  
}  
Global va lokal o'zgaruvchilarga murojaatni o'rganish  
// Muallif : Qudrat Abdurahimov  
// Sana : 05.11.2011  
// Maqsad : Global va lokal o'zgaruvchilarga murojaatni // o'rganish  
#include <iostream>  
int x = 5; // global o'zgaruvchi  
int main()  
{  
int x = 9; // lokal o'zgaruvchi  
std::cout << "lokal x=" << x << std::endl;  
std::cout << "global x=" << ::x << std::endl;  
return 0;  
}  
Kiritilgan n sonini 3 - darajasini hisoblovchi funksiya tuzilsin  
// Muallif : Qudrat Abdurahimov

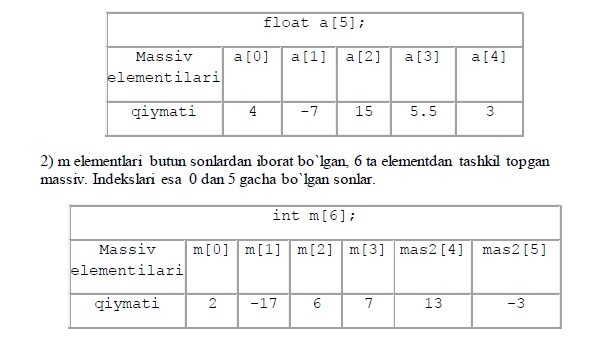
// Sana : 04.12.2011  
// Maqsad : Funksiyaga ko'rsatkich parametrlari  
// orqali murojaatni o'rganish  
#include <iostream>  
using namespace std;  
void kub (int \*);  
int main()  
{  
int n;  
cout << "n="; cin >> n;  
kub (&n);  
cout << "n ning qiymati =" << n << endl;  
return 0;  
}  
void kub (int \*nPtr)  
{  
\*nPtr = \*nPtr \* \*nPtr \* \*nPtr;  
}  
cout << "a="; cin >> a;  
cout << "b="; cin >> b;  
c = sum(a, b);  
cout << "1-sul natijasi=" << c << endl;  
sum(a, b, &c);  
cout << "2-sul natijasi=" << c << endl;  
c = sum(&a, &b);  
cout << "3-usul natijasi=" << c << endl;  
sum(&a, &b, &c);  
cout << "4-usul natijasi=" << c << endl;  
return 0;  
}  
// 1 - usul  
int sum(int son1, int son2)  
{  
int natija;  
natija = son1 + son2;  
return natija;  
}  
// 2 - usul  
void sum(int son1, int son2,int \*natija)  
{  
\*natija = son1 + son2;  
}  
// 3 - usul  
int sum(int \*son1, int \*son2)  
{  
int natija;  
natija = \*son1 + \*son2;  
return natija;  
}

// Sana : 04.12.2011  
// Maqsad : Funksiyaga ko'rsatkich parametrlari  
// orqali murojaatni o'rganish  
#include <iostream>  
using namespace std;  
void kub (int \*);  
int main()  
{  
int n;  
cout << "n="; cin >> n;  
kub (&n);  
cout << "n ning qiymati =" << n << endl;  
return 0;  
}  
void kub (int \*nPtr)  
{  
\*nPtr = \*nPtr \* \*nPtr \* \*nPtr;  
}  
Ikkita son yig'indisini funksiya orqali hisoblovchi programma tuzilsin  
// Muallif : Qudrat Abdurahimov  
// Sana : 04.12.2011  
// Maqsad : Funksiyaga qiymat va ko'rsatkich parametrlari  
// orqali murojaatni o'rganish  
#include <iostream>  
using namespace std;  
// funksiya prototipi  
int sum(int , int);  
void sum(int , int, int \*);  
int sum(int \*, int \*);  
void sum(int \*, int \*, int \*);  
int main()  
{  
int a, b, c;

// 4 - usul  
void sum(int \*son1, int \*son2,int \*natija)  
{  
\*natija = \*son1 + \*son2;  
}  
**Nazorat savollari:**  
1. Funksiya deb nimaga aytiladi?  
2. Parametrsiz funksiyaga murojaat qanday amalgam oshiriladi?  
3. Qanday o`zgaruvchilar global o`zgaruvchilar deyiladi?  
4. Qanday o`zgaruvchilar lokal o`zgaruvchilar deyiladi?  
5. Parametrli funksiya deb qanday funksiyalarga aytiladi?  
6. Qanday parametrlar qiymat parametrlar deyiladi?  
7. Qanday parametrlar ko'rsatkich parametrlar deyiladi?  
8. Funksiyadan chiqish uchun qaysi operatordan foydalaniladi?  
Ma’ruzalar matnining elektron versiyasidan mavzuga oid 60 ta masalani ishlab chiqing

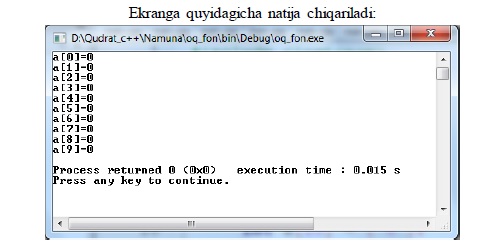
**Massivlar. Bir o`lchamli massivlar**

Massiv - bu bir xil toifali, chekli qiymatlarning tartiblangan to`plamidir. Massivlarga misol qilib matematika kursidan ma`lum bo`lgan vektorlar, matritsalarni ko`rsatish mumkin.  
Massiv bir o`lchamli deyiladi, agar uning elementiga bir indeks orqali murojaat qilish mumkin bo`lsa. Bir o`lchamli massivni e`lon qilish quyidagicha bo`ladi:  
< toifa> <massiv\_nomi> [ elementlar\_soni ] = { boshlang'ich qiymatlar }; Quyida massivlarni e`lon qilishga bir necha misollar keltirilgan:  
1) float a[5]; 2) int m[6]; 3) bool b[10];  
1) a elementlari haqiqiy sonlardan iborat bo`lgan, 5 ta elementdan tashkil topgan massiv. Indekslari esa 0 dan 4 gacha bo`lgan sonlar



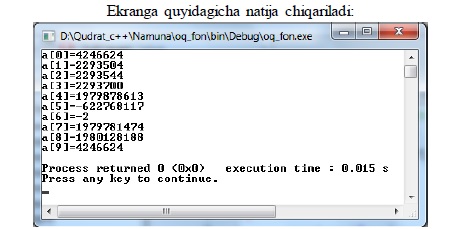
3) b elementlari mantiqiy qiymatlardan (true, false ) iborat bo`lgan 10 ta elementdan tashkil topgan massiv. Indekslari esa 0 dan 9 gacha bo`lgan sonlar.  
Massiv elementlariga murojaat qilish oddiy o`zgaruvchilarga murojaat qilishdan biroz farq qiladi. Massiv elementiga murojaat qilish uning indeksi orqali bo`ladi.  
a[1] = 10; a massivining 1 – elementi 10 qiymat o’zlashtirsin; cin >> a[2]; a massivining 2 – elementi kirtilsin; cout << a[3]; a massivining 3 – elementi ekranga chiqarilsin;  
Massivni e'lon qilishda uning elementlariga boshlang'ich qiymat berish mumkin va buning bir nechta usuli mavjud.  
1) O'lchami ko'rsatilgan massivni to'liq initsializatsiyalash. int k[5] = { 2, 3, 7, 8, 6};  
Bu yerda 5 ta elementdan iborat bo'lgan k massivi e'lon qilingan va massivning barcha elemantlariga boshlang'ich qiymat berilgan.  
2) O'lchami ko'rsatilgan massivni to'liqmas initsializatsiyalash. int k[5] = { 2, 3, 7 };  
Bu yerda 5 ta elementdan iborat bo'lgan k massivi e'lon qilingan va massivning dastlabki 3 ta elemantlariga boshlang'ich qiymat berilgan.

3) O'lchami ko'rsatilmagan massivni to'liq initsializatsiyalash. int k[] = { 2, 3, 7, 8, 6};  
Shuni takidlash lozimki, agar massiv o'lchami ko'rsatilmasa, uni to'liq initsializatsiyalash shart. Bu xolda massiv o'lchami kompilyatsiya jarayonida massiv elementlari soniga qarab aniqlanadi. Bu yerda massiv o'lchami 5 ga teng.  
4) O'lchami ko'rsatilgan massivning barcha elementlariga boshlang'ich qiymat 0 berish: int k[5] = { 0 };  
**O'lchami ko'rsatilgan massivning barcha elementlariga boshlang'ich qiymat 0 berish**  
#include <iostream>  
using namespace std;  
int main()  
{  
int a[10] = { 0 };  
//massivning barcha elementlariga 0 qiymat berish  
for (int i = 0; i < 10; i++)  
cout << "a[" << i << "]=" << a[i] << endl;  
return 0;  
}



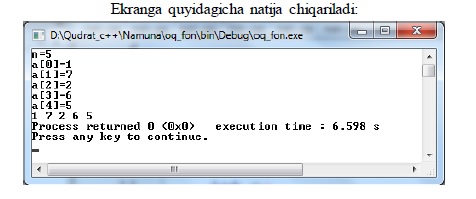
Agar massiv elementlariga boshlang'ich qiymatlar berilmasa xatolik sodir bo'lishi mumkin.

#include <iostream>  
using namespace std;  
int main()  
{  
int a[10];  
for (int i = 0; i < 10; i++)  
cout << "a[" << i << "]=" << a[i] << endl;  
return 0;  
}



Bunday natija chiqishining sababi video ma'ruzada batafsil tushuntiriladi.  
**Elementlari butun sonlardan iborat bo`lgan, n elementdan tashkil topgan massiv elementlarini kirituvchi va ekranga chiqaruvchi programma tuzilsin. ( n <= 10 )**

using namespace std;  
int main()  
{  
int a[10] = { 0 };  
int n;  
cout << "n="; cin >> n;  
for (int i = 0; i < n; i++)  
{  
cout << "a[" << i << "]=";  
cin >> a[i];  
}  
for (int i = 0; i < n; i++)  
cout << a[i] << " ";



**n ta elementdan tashkil topgan massiv berilgan. Shu massiv elementlari yig'indisini chiqatuvchi programma tuzilsin. ( n <= 10)**  
#include <iostream>  
using namespace std;  
int main()  
{  
int a[10] = { 0 }; // a massivini e'lon qilish  
int n; // massiv elamentlari soni  
int s = 0; // massiv elementlari yig'indisini hisoblash uchun  
cout << "n="; cin >> n;  
for (int i = 0; i < n; i++)  
{  
cout << "a[" << i << "]=";  
cin >> a[i];  
s += a[i];  
}  
cout << "Massiv elementlari yig`indisi = " << s << endl;  
return 0;  
}

**Massiv elementlari sonini quyidagicha aniqlash mumkin**  
Massivning kompyuter xotirasida egallagan hajmini, bitta elementi (massiv elementi toifasi) hajmiga bo'lish orqali.  
#include <iostream>  
using namespace std;  
int main()  
{  
int a[10];  
int n;  
cout << "n="; cin >> n;  
for (int i = 0; i < n; i++)  
{  
cout << "a[" << i << "]=";  
cin >> a[i];  
}  
cout << "1 - usul=" << sizeof(a) / sizeof(a[0]) << endl;  
cout << "2 - usul=" << sizeof(a) / sizeof(int) << endl;  
return 0;  
} **C stilida dinamik massivlar bilan ishlash**  
#include <iostream>  
#include <alloc.h>  
using namespace std;  
int main()  
{  
int n, s = 0;  
int \*a;  
cout << "n="; cin >> n;  
// xotira ajratish, xotira yetarli bo'lmasa  
// NULL (0) ko'rsatkich qaytariladi  
a =(int\*) malloc(n \* sizeof(int));

if ( a == NULL)  
{  
cout << "Xotira yetarli emas";  
return 1;  
}  
for (int i = 0; i < n; i++)  
{  
cout << "a[" << i << "]=";  
cin >> a[i];  
s += a[i];  
}  
// xotirani bo'shatish  
free(a);  
cout << s << endl;  
return 0;  
}  
**C++ stilida dinamik massivlar bilan ishlash**  
#include <iostream>  
using namespace std;  
int main()  
{  
int n, s = 0;  
int \*a;  
cout << "n="; cin >> n;  
// xotira ajratish, xotira yetarli bo'lmasa  
// NULL (0) ko'rsatkich qaytariladi  
a = new int[n];  
if ( a == NULL)  
{  
cout << "Xotira yetarli emas";  
return 1;  
}

for (int i = 0; i < n; i++)  
{  
cout << "a[" << i << "]=";  
cin >> a[i];  
s += a[i];  
}  
// xotirani bo'shatish  
delete []a;  
cout << s << endl;  
return 0;  
}  
**Nazorat savollari:**  
1. Massiv nima?  
2. Bir o'lchamli massiv deb nimaga aytiladi?  
3. Massiv elementi nima?  
4. Massiv indeksi nima?  
5. Bir o'lchamli massiv qanday e`lon qilinadi?  
6. Massiv elementlari soni qanday aniqlanadi?  
Ma’ruzalar matnining elektron versiyasidan mavzuga oid 140 ta masalani ishlab chiqing.