

15-16-AMALIY MASHG'ULOT. FUNKSIYALAR VA ULARNI E'LON QILISHGA OID DASTURLAR TUZISH

4.1. “Lokal va global”. O'zgaruvchilarni lokal va global ko'rinishda e'lon qiling. Agar ularning nomlari bir xil bo'lib qolganda qo'llaniladigan usulni keltiring.

```
#include <iostream>
int a = 1; // Global
int c = 10; // Global
using namespace std;
void main()
{
    int b = 1; // Lokal
    float c = 4.3; // Lokal
    cout << "b=" << b << endl;
    cout << "a=" << a << endl;
    cout << "Global c=" << ::c << endl; // Global o'zgaruvchiga ruxsat berish;
    cout << "Lokal c=" << c << endl;
}
```

4.2. “Nomlar fazosi”. Shaxsiy nomlar fazosini yarating. Unda ikkita funksiya va 5 ta o'zgaruvchi e'lon qilib, ulardan dasturning asosiy tanasida foydalaning.

```
#include <iostream>
#include <math.h>
using namespace std;
namespace Erali
{
    int a, b, c;
    char d;
    float k;
    void Namoyish()
    {
        for (int i = 1; i <= 20; i++)
            cout << i << " ";
    };
    void Tezlik()
    {
```

```

        int i = 10;
        while (i < 100)
        {
            i += 5;
            cout << i << " ";
        }
    }

void main()
{
    Erali::a = 10;
    cout << "a=" << Erali::a << endl;
    Erali::Namoyish();
    Erali::Tezlik();
    return;
}

```

4.3. “Maks son”. a va b sonlardan eng kattasini topishni funksiya yordamida hal eting.

```

#include <iostream>

using namespace std;
int Maximum(int a, int b)
{
    if (a>b)
        return a;
    else
        return b;
    cout << a;
}

int main()
{
    int a1, b1, c;
    cout << "a="; cin >> a1;
    cout << "b="; cin >> b1;
    c = Maximum(a1, b1);
    cout << a1 << " va " << b1 << " sonlaridan kattasi " << c;
    return 0;
}

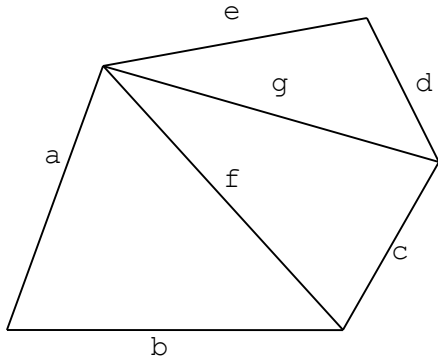
```

```
}
```

4.4. “Fakt yig’indi”. $1!+2!+\dots+n!$ йиғиндини ҳисобланг (n нинг қиймати клавиатурадан киритилади; $1 < n \leq 10$).

```
#include <iostream>
#include <math.h>
using namespace std;
long Fakt_yigindi(int n, long S)
{
    S = 1;
    for (int i = 1; i <= n; i++)
        S *= i;
    return S;
}
int main()
{
    int n, Y = 0;
    cout << " n ni kiriting" << endl;
    cin >> n;
    for (int i = 1; i <= n; i++)
        Y += Fakt_yigindi(i, Y);
    cout << "Natija S=" << Y;
    return 0;
}
```

4.5. “Yuza”. a,b,c,d,e,f,g ҳақиқий сонлар берилган. Расмдаги тавсифланган бешбурчакнинг юзасини топинг. Учбурчак томонларига кўра юзасини ҳисоблаш учун функция аниқланг.



```
#include <iostream>
#include <math.h>
using namespace std;
```

```
float Yuza(float a, float b, float c)
{
    double p, S;
    p = (a + b + c) / 2;
    S = sqrt(p*(p - a)*(p - b)*(p - c));
    return S;
}
int main()
{
    float a, b, c;
    float S, S1=0;
    for (int i = 1; i <= 3; i++) //Funksiyadan 3 marta foydalanish
    {
        cin >> a >> b >> c;
        S = Yuza(a, b, c);
        S1 += S;
    }
    cout << "Yuzasi S=" << S << endl;
    return 0;
}
```

4.6. “Tub son”. Tub sonlarni aniqlashga imkon beruvchi funksiyani aniqlab, barcha uch xonali tub sonlarni aniqlang.

```
#include <iostream>
```

```

using namespace std;
int Tub_son(int i)
{
    int b = 0, S = 0;
    for (int j = 1; j <= i; j++)
        if (i%j == 0)
            S++;
    if (S == 2)
        b = 1;
    return b;
}

int main()
{
    int b;
    cout << "Barcha uch xonali tub sonlar quyidagilardir:" << endl;
    for (int i = 100; i <= 999; i++)
    {
        b = Tub_son(i);
        if (b == 1)
            cout << i << endl;
    }
    return 0;
}

```

4.7. “Ketma-ketlik”. Butun sonlarning ikkita ketma-ketmaligi berilgan: a_1, a_2, \dots, a_8 va b_1, b_2, \dots, b_8 . Juft sonlarni aniqlashga ruxsat beradigan funksiyani aniqlab, ulardan birinchisining juft sonlari sonini va ikkinchisining toq sonlari sonini toping.

```

#include <iostream>

```

```

using namespace std;

```

```

int Son(int a)
{
    if (a % 2 == 0)
        return 1;
    else
        return 0;
}

```

```

}
int main()
{
    int a, k, S=0, S1;
    for (int j = 1; j <= 2; j++)
    {
        cout << j << "-qator:";
        for (int i = 1; i <= 8; i++)
        {
            cin >> a;
            k = Son(a);
            if (k)
                S++;
        }
        cout << j << "-qatordagi juft sonlar soni:" << S << endl;
        S1 = S;
        S = 0;
    }
    cout << 2 << "-qatordagi toq sonlar soni:" << 8-S1 << endl;
    return 0;
}

```

4.8. “Tub son”. $(n, 2n)$ oralig’idagi tub sonlarni topish funksiyasini yozing.

```

#include <iostream>

using namespace std;
bool Tub_son(int n)
{
    unsigned int S=0;
    for (int j = 2; j < n / 2; j++)
        if (n%j == 0)
            S++;
    if (S == 0)
        return 1;
    else
        return 0;
}
int main()
{

```

```

    unsigned int n;
    bool k;
    cout << "n="; cin >> n;
    for (int i = n; i <= 2 * n; i++)
    {
        k = Tub_son(i);
        if (k)
            cout << i << " ";
    }
    return 0;
}

```

4.9. “Egizak son”. Agar ikkita tub sonlar bir-biridan ikkiga farq qilsa, ular “egizaklar” deyiladi. Masalan, 41 va 43. Barcha 200 dan oshmaydigan egizak tub sonlar juftliklarini topuvchi funksiyani yozing.

```

#include <iostream>

using namespace std;
bool Tub_son(int n)
{
    unsigned int S=0;
    for (int j = 2; j < n / 2; j++)
        if (n%j == 0)
            S++;
    if (S == 0)
        return 1;
    else
        return 0;
}

void main()
{
    unsigned int a, b=0, n;
    bool k;
    for (int i = 1; i <= 200; i++)
    {
        k = Tub_son(i);
        if (k)
        {
            a = i;

```

```

        if ( a-b == 2)
        {
            cout << a << "," << b << " ";
        }
        b = a;
    }
}
return;
}

```

4.10. “Baxtli son”. Ҳамма олтихонали бахтли сонларни олинг. Олтихонали сонларнинг биринчи учта рақамлари йиғиндиси, охирги учта рақамлар йиғиндисига тенг бўлса, бу сон бахтли ҳисобланади. Уч хонали сонларнинг рақамлар йиғиндисини ҳисоблаш учун функция аниқланг.

```

#include <iostream>

```

```

using namespace std;

```

```

bool Baxtli_Son(unsigned int n)

```

```

{
    unsigned int n1, n2;
    n1 = n / 1000;
    n2 = n % 1000;
    if (n1 == n2)
        return 1;
    else
        return 0;
}

```

```

void main()

```

```

{
    bool k;
    cout << "Baxtli sonlar ro'yxati:" << endl;
    for (int i = 100000; i <= 999999; i++)
    {
        k = Baxtli_Son(i);
        if (k)
            cout << i << endl;
    }
}

```



```

    }
    return;
}

```

4.11. “Faktorial funksiya”. Натурал сон факториалини ҳисоблайдиган функцияни аниқлаб $\frac{2 \cdot 5! + 3 \cdot 8!}{6! + 4!}$ ифоданинг қийматини топинг.

```

#include <iostream>

```

```

using namespace std;

```

```

int fakt(int n)
{
    int S = 1;
    for (int i = 1; i <= n; i++)
        S *= i;
    return S;
}

```

```

int main()
{
    float x;
    x = (2 * fakt(5) + 3 * fakt(8)) / (fakt(6) + fakt(4));
    cout << "x=" << x;
    return 0;
}

```

4.12. “EKUB”. EKUB(a,b,c) ni aniqlang.

```

#include <iostream>

```

```

using namespace std;

```

```

int EKUB(unsigned int a, unsigned int b)
{
    while (a != b)

```

```

    {
        if (a > b)
            a = a - b;
        else
            b = b - a;
    }
    return a;
}

void main()
{
    int EKUB1, a, b, c;
    cin >> a >> b >> c;
    EKUB1 = EKUB(c, EKUB(a, b));
    cout << EKUB1;
    return;
}

```

4.13. “EKUB”. n ta sonning EKUBini toping

```

#include <iostream>

using namespace std;

int EKUB(unsigned int a, unsigned int b)
{
    while (a != b)
    {
        if (a > b)
            a = a - b;
        else
            b = b - a;
    }
    return a;
}

void main()
{
    int EKUB1, a, b, c, n;

```

```
cin >> n;
cout << "a="; cin >> a;
cout << "b="; cin >> b;
for (int i = 0; i < n-2; i++)
{
    EKUB1 = EKUB(a, b);
    cout << "a="; cin >> a;
    b = EKUB1;
}
cout << "EKUB=" << EKUB1;
return;
}
```