

Име.....Фак. номер.....

**Изпит по увод в програмирането на базата на езика C++, 22.02.2013**

**Вариант 1**

**Задача 1 (4 точки).** Да се запишат на езика C++ следните изрази:

$$\text{a)} \frac{a}{b \cdot \frac{c}{d \cdot \frac{e}{f \cdot h}}} \quad \text{б)} \frac{a + b}{x - 2 \cdot y} \quad \text{в)} a + \frac{b}{x - 2} \cdot y$$

$$\text{г)} \frac{(\log_3 |x-2| + e^{\frac{x-y}{2}})^3}{\lg(2 + e^{\frac{x+y}{2}})}$$

а)  $a / (b * (c / (d * (e / (f * h)))))$

б)  $(a+b) / (x - 2*y)$

в)  $a + (b / (x - 2)) * y$

г)  $\text{pow}((\log(\text{abs}(x-2))/\log 3 + \exp((x-y)/2)), 3) / \log 10(2 + \exp((x+y)/2))$

**Задача 2 (3 точки).** Да се опростят булевите изрази като се приложи операцията !:

а)  $!(a > 0 \ \&\& \ a < 7) - a \leq 0 \ || \ a \geq 7$

б)  $!(!(a \geq 0) \ || \ (a \geq 7 \ \&\& \ a \leq 10)) - (a \geq 0) \ \&\& \ (a < 7 \ || \ a > 10)$

в)  $!(a \geq -4 \ \&\& \ a \leq -2 \ || \ !(a < 2) \ \&\& \ a \leq 4) - a < -4 \ || \ a > -2 \ \&\& \ a \geq 2 \ || \ a > 4$

**Задача 3 (3 точки).** Дадени са точките  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$ . Да се напише програмен фрагмент, който определя дали точките могат да са върхове на триъгълник.

```
double x1, y1, x2, y2, x3, y3;
double a = sqrt(pow(x1-x2, 2) + pow(y1-y2, 2));
double b = sqrt(pow(x2-x3, 2) + pow(y2-y3, 2));
double c = sqrt(pow(x1-x3, 2) + pow(y1-y3, 2));
bool isTriangle = (a < b + c && b < a + c && c < b + a);
if(isTriangle) {
    cout << "Points form a triangle.";
}
```

```

} else {
    cout << "Points don't form a triangle.";
}

```

**Задача 4 (4 точки).** Да се напише програмен фрагмент, който изследва за решение системата уравнения:

$$\begin{cases} a_1 x + b_1 y = c_1 \\ a_2 x + b_2 y = c_2 \end{cases}$$

Намира решение на системата, ако такова съществува ( $a_1$ ,  $b_1$ ,  $c_1$ ,  $a_2$ ,  $b_2$  и  $c_2$  са дадени реални числа).

```

#include<iostream>
#include<iomanip>
#include<cmath>
using namespace std;
int main()
{
    cout.precision(4);
    cout.setf(ios::fixed);
    int n,i,j,k,flag=0,count=0;
    cout<<"\nEnter the no. of equations\n";
    cin>>n;          //Input no. of equations
    double a[n][n+1];    //declare a 2d array for storing the elements of the augmented
matrix
    double x[n];          //declare an array to store the values of variables
    double eps,y;
    cout<<"\nEnter the elements of the augmented matrix row-wise:\n";
    for (i=0;i<n;i++)
        for (j=0;j<=n;j++)
            cin>>a[i][j];
    cout<<"\nEnter the initial values of the variables:\n";
    for (i=0;i<n;i++)
        cin>>x[i];
    cout<<"\nEnter the accuracy upto which you want the solution:\n";
    cin>>eps;
    for (i=0;i<n;i++)    //Pivotisation(partial) to make the equations diagonally dominant
        for (k=i+1;k<n;k++)
            if (abs(a[i][i])<abs(a[k][i]))
                for (j=0;j<=n;j++)

```

```

        {
            double temp=a[i][j];
            a[i][j]=a[k][j];
            a[k][j]=temp;
        }
    cout<<"Iter"<<setw(10);
    for(i=0;i<n;i++)
        cout<<"x"<<i<<setw(18);
    cout<<"\n-----";
    do
        //Perform iterations to calculate x1,x2,...xn
    {
        cout<<"\n"<<count+1<<". "<<setw(16);
        for (i=0;i<n;i++)
            //Loop that calculates x1,x2,...xn
        {
            y=x[i];
            x[i]=a[i][n];
            for (j=0;j<n;j++)
            {
                if (j!=i)
                    x[i]=x[i]-a[i][j]*x[j];
            }
            x[i]=x[i]/a[i][i];
            if (abs(x[i]-y)<=eps)
                //Compare the ne value with the last value
                flag++;
            cout<<x[i]<<setw(18);
        }
        cout<<"\n";
        count++;
    }while(flag<n);
    //If the values of all the variables don't differ from their
    previous values with error more than eps then flag must be n and hence stop the loop

    cout<<"\n The solution is as follows:\n";
    for (i=0;i<n;i++)
        cout<<"x"<<i<<" = "<<x[i]<<endl;
    //Print the contents of x[]
    return 0;
}

```

**Задача 5 (4 точки).** Да се напише програмен фрагмент, който намира стойността на верижната дроб ( $x$  е дадено реално число):

$$\frac{x}{x^2 + \frac{2}{x^2 + \frac{4}{x^2 + \frac{8}{\ddots x^2 + \frac{256}{x^2}}}}}.$$

```
double x;
double result = x*x;
for(size_t i = 256; i >= 2; i = i / 2) {
    double helper = i / result;
    result = x*x + helper;
}
result = x / result;
```

**Задача 6 (4 точки).** Да се напише програмен фрагмент, който проверява дали редицата от реални числа  $a_0, a_1, \dots, a_{n-1}$  ( $1 \leq n \leq 100$ ) е трион от вида:  $a_0 > a_1 < a_2 > \dots a_{n-1}$ .

```
int n;
double* a = new double[n];
bool isTrion = true;
for(size_t i = 0; i < n; i+=2) {
    if(i % 2 == 1) {
        if(i==n-1) {
            break;
        }
        if(!(a[i] < a[i-1] && a[i] < a[i+1])) {
            isTrion = false;
            break;
        }
    }
}
if(isTrion) {
    cout << "Sequence is trion.";
} else {
    cout << "Sequence is not trion.";
}
```

**Задача 7 (6 точки).** Дадени са две редици от числа. Да се напише програмен фрагмент, който определя колко пъти първата редица се съдържа във втората. Например редицата 1, 2, 3 се съдържа 2 пъти в редицата 3, 4, 1, 2, 3, 5, 6, 1, 2, 5, 3, 8, 1, 2, 3, 4.

```

int n;
int* a = new int[n];
int m;
int* b = new int[m];
//cin ...
int occurenceCounter = 0;
for(size_t i = 0; i < m; i++) {
    if(b[i]==a[0]) {
        bool isFullyContained = true;
        i++;
        for(size_t j = 1; j < n; j++) {
            if(a[j] == b[j]) {
                i++;
                continue;
            } else {
                isFullyContainde = false;
                break;
            }
        }
        occurenceCounter += isFullyContained;
    }
}
cout << isFullyContained;

```

**Задача 8 (4 точки).** Какъв е резултатът от изпълнението на програмата?

<pre> #include &lt;iostream&gt; using namespace std;  void func(int x, int&amp; y, int* z) { int a = 3;   int b = 4;   a = b + y;   y = x + a;   z = &amp;a;   cout &lt;&lt; "func: x = " &lt;&lt; x &lt;&lt; endl;   cout &lt;&lt; "func: y = " &lt;&lt; y &lt;&lt; endl;   cout &lt;&lt; "func: *z = " &lt;&lt; *z&lt;&lt; endl;   cout &lt;&lt; "func: a = " &lt;&lt; a &lt;&lt; endl;   cout &lt;&lt; "func: b = " &lt;&lt; b &lt;&lt; endl;   return; }  int main() { int a = 1; </pre>	<pre> func: x = 3 func: y = 9 func: *z = 6 func: a = 6 func: b = 4 main: a = 1 main: b = 9 </pre>
--	---

<pre> int b = 2; func(a+b, b, &amp;a); cout &lt;&lt; "main: a = " &lt;&lt; a &lt;&lt; endl; cout &lt;&lt; "main: b = " &lt;&lt; b &lt;&lt; endl; return 0; } </pre>	
---	--

**Задача 9 (4 точки).** Да се напише булева функция, която проверява дали число е степен на 5.

Има и мн. други начини, но напр.:

```

bool isPowOfFive(int n) {
    if(n % 5 == 1 && n == 1) {
        return true;
    }
    if(n%5 != 0) {
        return false;
    }
    return isPowOfFive(n/5);
}

```

**Задача 10 (4 точки).** Да се напише булева функция, която проверява дали редица от числа съдържа число, което е степен на 5. За целта да се използва функцията от задача 9.

```

//нека редицата е arr, с дължина n
bool containsSuch(int* arr, int n) {

    for(int i = 0; i < n; i++) {
        if(isPowOfFive(arr[i])) {
            return true;
        }
    }

    return false; //може и без този ред;
}

```

**Задача 11 (6 точки).** Да се дефинира функция, която извежда на екрана елементите на квадратна матрица като ги обхожда по диагонали, успоредни на вторичния главен диагонал, започвайки от долния десен ъгъл.

```

void print(int** matrix, int n) {
    for (int i = 2 * (n - 1); i >= 0; i--) {
        for (int j = n - 1; j >= 0; j--) {
            for (int k = 0; k < n; k++) {
                if (j + k == i) {

```

```

        cout << matrix[j][k] << " ";
    }
}
}
cout << endl;
}
}

```

//за да си тествате:

```

int main(int argc, char** argv)
{
    int** matrix = new int* [4];
    for (int i = 0; i < 4; i++) {
        matrix[i] = new int[4];
    }
    for (int j = 0; j < 4; j++) {
        for (int k = 0; k < 4; k++) {
            cin >> matrix[j][k];
        }
    }
    print(matrix, 4);
}

```

**Задача 12 (6 точки).** Да се дефинира рекурсивна функция, която проверява дали елементите на редица съдържат даден елемент.

```

bool contains(int* arr, int n, int element) {
    if (n == 1) {
        return arr[0] == element;
    }
    return arr[n - 1] == element || contains(arr, n - 1, element);
}

```

```

int main(int argc, char** argv)
{
    int* arr = new int[5]{ 3, 2, 5, 23, 0 };
    cout << contains(arr, 5, 5);
    cout << contains(arr, 5, 6);
}

```

Име.....Фак. номер.....

**Изпит по увод в програмирането на базата на езика C++, 22.02.2013**

**Вариант 2**

**Задача 1 (4 точки).** Да се запишат на езика C++ следните изрази:

а)  $\frac{a+b}{b-d} \cdot x$

б)  $\sqrt{\left| a \cdot \frac{b}{c-d} \right|}$

в)  $\frac{a+b}{b+\frac{b+c}{c+\frac{d}{e+f}}}$

а)  $((a+b)/(b-d)) * x$

б)  $\text{sqrt}(\text{abs}(a*(b/(c-d))))$

в)  $(a+b)/(b + ((b+c)/(c + b/(e+f))))$

г)  $\text{atan}(x) + a/\tan(x) - (\text{pow}(\sin(x*x) + \cos(x*x*x), 2))/(\log_{10}(\text{abs}(2+x)))$

г)  $\text{arctg } x + \text{cotg } x - \frac{(\sin x^2 + \cos x^3)^2}{\lg|2 + x|} \dots\dots\dots$

**Задача 2 (3 точки).** Да се опростят булевите изрази като се приложи операцията !:

а)  $!(a > 0 \ || \ a < 7)$

б)  $!(!(a > 0) \ \&\& \ (a < -7 \ || \ a > -3))$

в)  $!(a > -6 \ \&\& \ a < -4 \ || \ !(a \leq 4) \ \&\& \ a < 6)$

- Като по дстр

**Задача 3 (3 точки).** Известно е, че точките  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$  са върхове на триъгълник. Да се напише програмен фрагмент, който определя вида на триъгълника – равностранен, равнобедрен, разностранен.

```
double x1, y1, x2, y2, x3, y3;
double a = sqrt(pow(x1-x2, 2) + pow(y1-y2, 2));
double b = sqrt(pow(x2-x3, 2) + pow(y2-y3, 2));
double c = sqrt(pow(x1-x3, 2) + pow(y1-y3, 2));
if(a==b&& a==c) {
    cout << "Равностранен";
} else if (a==b && a != c) {
    cout << "Равнобедрен";
} else {
```



```

        cout << "Разностранен";
    }

```

**Задача 4 (4 точки).** Да се напише програмен фрагмент, който изследва за решение уравнението:

$$|x-3| + |x-5| = 2$$

Намира и извежда решение, ако такова съществува ( $x$  е реална променлива).

**Задача 5 (4 точки).** Да се напише програмен фрагмент, който намира стойността на верижната дроб:

$$1 + \frac{1}{3 + \frac{1}{5 + \frac{1}{\ddots + \frac{1}{109 + \frac{1}{111}}}}}$$

```

double result = 111;
for(size_t i = 111; i >= 1; i-=2) {
    double helper = 1/result;
    result = i + helper;
}

```

**Задача 6 (4 точки).** Да се напише програмен фрагмент, който проверява дали редицата от реални числа  $a_0, a_1, \dots, a_{n-1}$  ( $1 \leq n \leq 100$ ) е трион от вида:  $a_0 < a_1 > a_2 < \dots a_{n-1}$ .

```

int n;
double* a = new double[n];
bool isTrion = true;
for(size_t i = 0; i < n; i+=2) {
    if(i % 2 == 1) {
        if(i==n-1) {
            break;
        }
        if(!(a[i] > a[i-1] && a[i] > a[i+1])) {
            isTrion = false;
            break;
        }
    }
}

```

```

    }
}
if(isTrion) {
    cout << "Sequence is trion.";
} else {
    cout << "Sequence is not trion.";
}

```

**Задача 7 (6 точки).** Дадени са две редици от числа. Те представят две множества. Да се напише програмен фрагмент, който определя дали първото множество е подмножество на второто.

//декларация, въвеждане; нека дължината на първия масив a (той представя първото множество) е n, на втория масив b (второто множество) – m.

```

bool isSubset = true;
for(size_t i = 0; i < n; i++) {
    bool currentIsFound = false;
    for(size_t j = 0; j < m; j++) {
        if(a[i] == b[j]) {
            currentIsFound = true;
            break;
        }
    }
    if(!currentIsFound) {
        isSubset = false;
        break;
    }
}

```

**Задача 8 (4 точки).** Какъв е резултатът от изпълнението на програмата?

<pre> #include &lt;iostream&gt; using namespace std;  void func(int x, int&amp; y, int* z) { int a = 7;   int b = 1;   b = a + y;   y = x + a;   z = &amp;b;   cout &lt;&lt; "func: x = " &lt;&lt; x &lt;&lt; endl;   cout &lt;&lt; "func: y = " &lt;&lt; y &lt;&lt; endl;   cout &lt;&lt; "func: *z = " &lt;&lt; *z </pre>	<pre> func: x = -2 func: y = 5 func: *z = 11 func: a = 7 func: b = 11 main: a = 5 main: b = 6 </pre>
---	--

<pre>         &lt;&lt; endl;         cout &lt;&lt; "func: a = " &lt;&lt; a &lt;&lt; endl;         cout &lt;&lt; "func: b = " &lt;&lt; b &lt;&lt; endl;         return;     }     int main()     { int a = 4;       int b = 6;       func(a-b, a, &amp;b);       cout &lt;&lt; "main: a = " &lt;&lt; a &lt;&lt; endl;       cout &lt;&lt; "main: b = " &lt;&lt; b &lt;&lt; endl;       return 0;     } </pre>	
--	--

**Задача 9 (4 точки).** Да се напише булева функция, която проверява дали естествено число е просто.

```

bool isPrime(int n) {
    for(size_t i=2; i <= sqrt(n); i++) {
        if(n%i==0) {
            return true;
        }
    }
    return false; //може и без този ред
}

```

**Задача 10 (4 точки).** Да се напише булева функция, която проверява дали редица от числа съдържа число, което е просто. За целта да се използва функцията, дефинирана в задача 9.

```

//нека редицата е arr, с дължина n
bool containsSuch(int* arr, int n) {
    for(int i = 0; i < n; i++) {
        if(isPrime(arr[i])) {
            return true;
        }
    }
    return false; //може и без този ред;
}

```

**Задача 11 (6 точки).** Да се дефинира функция, която извежда на екрана елементите на квадратна матрица като ги обхожда по диагонали, успоредни на главния диагонал, започвайки от долния ляв ъгъл.

```
void print(int** matrix, int n) {
    for (int k = n - 1; k >= -(n - 1); k--) {
        for (int i = n - 1; i >= 0; i--) {
            for (int j = 0; j < n; j++) {
                if (i - j == k) {
                    cout << matrix[i][j] << " ";
                }
            }
        }
        cout << endl;
    }
}
```

**Задача 12 (6 точки).** Да се дефинира рекурсивна функция, която проверява дали редица от числа съдържа елемент, който е принадлежи на интервала [a, b].

```
bool contains(int* arr, int n, int a, int b) {
    if (n == 1) {
        return arr[0] >= a && arr[0] <= b;
    }
    return arr[n - 1] >= a && arr[n - 1] <= b || contains(arr, n - 1, a, b);
}
```