

Изпит по увод в програмирането на базата на езика C++, 17.02.2014
Вариант 1

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

$$\begin{array}{ll} \text{a) } \sin\left(\frac{x}{y \cdot z}\right) + \frac{(\operatorname{tg}^2 x + \cos^2 x)^3}{\lg(1 + x^4)} & \text{б) } \frac{(\ln|x-1| + e^{\frac{x-y}{2}})^2}{\ln(2 + e^{\frac{x+y}{2}})} \cdot \frac{1}{y} \\ \text{в) } \frac{\sqrt[2]{x} + \sqrt[3]{y} + \sqrt[4]{z}}{|x+y+z|} & \text{г) } \frac{-b + \sqrt{|b^3 - c^3|}}{a+2} \end{array}$$

- a) $\sin(x/(y*z)) + (\operatorname{pow}((\operatorname{tg}(x)*\operatorname{tg}(x) + \cos(x)*\cos(x), 3))/(\log10(1+\operatorname{pow}(x,4))))$
 б) $(\operatorname{pow}(\log(\operatorname{abs}(x-1)) + \exp((x-y)/2), 2)/(\log(2+\exp((x+y)/2))))*((\operatorname{double})1/y)$
 в) $(\operatorname{sqrt}(2) + \operatorname{pow}(y, (\operatorname{double})1 / 3) + \operatorname{pow}(z, (\operatorname{double})1 / 4))/\operatorname{abs}(x+y+z)$
 г) $(-b + \operatorname{sqrt}(\operatorname{abs}(b*b*b - c*c*c)))/(a+2)$

Задача 2 (4 точки). Дадени са целите числа a , b и c .

а) Да се дефинира булев израз, който има стойност *true*, ако съществува такова разместване на числата, при което те образуват аритметична прогресия и има стойност *false*, в противен случай.

$(a \geq b \ \&\& \ b \geq c) \ || \ (b \geq a \ \&\& \ a \geq c) \ || \ (c \geq a \ \&\& \ a \geq b) \ || \ (a \geq c \ \&\& \ c \geq b) \ || \ (b \geq c \ \&\& \ c \geq a) \ || \ (c \geq b \ \&\& \ b \geq a);$

б) Да се намери отрицанието на булевия израз, дефиниран в а).

опростявате $!(a \geq b \ \&\& \ b \geq c) \ || \ (b \geq a \ \&\& \ a \geq c) \ || \ (c \geq a \ \&\& \ a \geq b) \ || \ (a \geq c \ \&\& \ c \geq b) \ || \ (b \geq c \ \&\& \ c \geq a) \ || \ (c \geq b \ \&\& \ b \geq a)$ като по дстр;

Задача 3 (5 точки). Дадени са целите числа a , b , c и d . Да се напише програма, която извежда на екрана средно-аритметичното на четните числа сред дадените. Ако няма четни числа, на екрана да се изведе съобщението „няма четни числа”.

```
int sum = 0;
int countEven = 0;
if(a%2==0) {
    countEven++;
    sum += a;
}
if(b%2==0) {
    countEven++;
```

```

        sum += b;
    }
    if(c%2==0) {
        countEven++;
        sum += c;
    }
    if(d%2==0) {
        countEven++;
        sum += d;
    }

    double avg = (double)sum / countEven;

```

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

$$S = \frac{2}{3 \cdot 4!} + \frac{4}{5 \cdot 6!} + \dots + \frac{2n}{(2n+1)(2n+2)!}$$

където n е дадено естествено число.

```

int factorial(int n) {
    if(n == 0) {
        return 1;
    }
    return n*factorial(n-1);
}

...
double sum = 0;
int n;
for(int i = 2; i<=2*n; i++) {
    sum += 2 / ((i+1)*factorial(i+2));
}

```

Задача 5 (6 точки). Дадена е редица от цели числа, която съдържа точно веднъж числото -1 и има поне 2 елемента преди -1. Да се дефинира програмен фрагмент, който проверява дали в частта на редицата, започваща от първия елемент до елемента, равен на -1 има два последователни равни елемента. В случай, че има, да се изведат поредните номера на числата на първата такава двойка елементи. При реализиране на търсенето редицата да се обходи еднократно като се използва операторът за цикъл *do-while*.

Задача 6 (6 точки). Даден е масив a от $2n$ цели числа ($n > 0$ е дадено цяло число). Да се напише функция, която връща *true*, ако всяко число от първата половина на a е по-голямо от всяко число от втората половина на a . За целта да се формулира и релизира подходяща задача за съществуване. Да не се влагат *return* и *break* в операторите за цикъл.

Задача 7 (4 точки). Какъв е резултатът от изпълнението на програмата?

<pre> #include <iostream> using namespace std; void f(int x, int& y, int* z) { int p = 1, q = 2; p = (q + y)/2; y = x + p; z = &p; cout << "f: x = " << x << endl; cout << "f: y = " << y << endl; cout << "f: *z = " << *z << endl; cout << "f: p = " << p << endl; cout << "f: q = " << q << endl; return; } int main() { int p = 4, &q = p; f(p+q, q, &p); cout << "main: p = " << p << endl; cout << "main: q = " << q << endl; return 0; } </pre>	<pre> f: x = 8 f: y = 11 f: *z = 3 f: p = 3 f: q = 2 main: p = 11 main: q = 11 </pre>
--	---

Задача 8 (4 точки). Дадени са две редици от реални числа: a с n елемента и сортирана във възходящ ред и b с m елемента и сортирана в низходящ ред. Да се дефинира функция,

която слива редиците a и b в нова редица c с $n+m$ елемента и сортирана във възходящ ред. За целта да се модифицира изветният алгоритъм за сливане *merge*.

//за с сме си заделили памет $n+m$ там където викаме тази функция

```
void merge(int* a, int n, int* b, int m, int* c) {
    int indexA = n-1, indexB = n-1, indexC = 0;
    while(indexA >= 0 && indexB >= 0) {
        if(a[indexA] > b[indexB]) {
            c[indexC] = a[indexA];
            indexA--;
        } else {
            c[indexC] = b[indexB];
            indexB--;
        }
        indexC++;
    }
    if(indexA != 0) {
        while(indexA >= 0) {
            c[indexC] = a[indexA];
            indexA--;
        }
    }
    if(indexB != 0) {
        while(indexB >= 0) {
            c[indexC] = b[indexB];
            indexB--;
        }
    }
}
```

Задача 9 (5 точки). Дадена е квадратна матрица a от порядък n от цели числа. Да се напише булева функция, която връща *true*, ако в a има ред, всички елементи на който са удвоени нечетни числа.

```
bool isDoubledOdd(int n) {
    return (n/2)%2 == 1;
}
```

```
bool allElementsAreDoubledOdd(int* arr, int n) {
    for(size_t i = 0; i < n; i++) {
        if(!isDoubledOdd(arr[i])) {
            return false;
        }
    }
    return true; //тук не може без този ред!
}
```

```
bool hasSuchRow(int** matrix, int n) {
    for(size_t i = 0; i < n; i++) {
        if(allElementsAreDoubledOdd(matrix[i], n)) {
```

```

        return true;
    }
}
return false; //тук обаче може.
}

```

Задача 10 (6 точки). Дадена е редица от числа a . Да се дефинира функция от по-висок ред, която прилага реалната едноаргументна числова функция f над елементите на a . Обхождането на редицата a е от края към началото ѝ.

```

double* map(T* arr, int size, double(*f)(double))
{
    double* newArr = new double[size];
    for (size_t i = size-1; i >= 0; i--)
    {
        newArr[i] = f(arr[i]);
    }
    return newArr;
}

```

Задача 11 (5 точки). Дадени са първият член и разликата на аритметична прогресия. Да се дефинират рекурсивни функции за намиране на:

- а) n -тия член на прогресията;
- б) сумата от първите n члена на прогресията.

а)

```

double progressionRec(double a1, double d, double n) {
    if(n <= 1){
        return a1;
    } else {
        double next = a1 + d;
        return progressionRec(next, d, n - 1);
    }
}

```

б)

```

double progressionRec(double a1, double d, double n) {
    if(n <= 1){
        return a1;
    } else {
        double next = a1 + d;
        return a1 + progressionRec(next, d, n - 1);
    }
}

```

Задача 12 (6 точки). Даден е масив a от $2n$ цели числа ($n > 0$ е дадено цяло число). Да се напише рекурсивна функция, която връща *true*, ако всяко число в първата половина на a е по-голямо от всяко число във втората половина на a .

Задача 13 (8 точки). Да се дефинира рекурсивна функция, която въвежда от клавиатурата, записан без грешка, израз от вида:

$$\langle \text{израз} \rangle ::= f(\langle \text{израз} \rangle, \langle \text{израз} \rangle) \mid g(\langle \text{израз} \rangle, \langle \text{израз} \rangle) \mid \langle \text{цифра} \rangle$$

където f и g са цели функции, дефинирани по следния начин:

$$f(x, y) = (x+y)\%3,$$

$$g(x, y) = |x-y|$$

Функцията да намира стойността на израза.

Име.....Фак. номер.....

Изпит по увод в програмирането на базата на езика C++, 17.02.2014

Вариант 2

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

a) $\cos\left(\frac{x}{y \cdot z}\right) + \frac{(\cot^2 x + \sin^2 x)^3}{\lg(1 + x^2)}$

б) $\frac{(\ln|x-1| + e^{\frac{x+y}{2}})^3}{\ln|3 - e^{\frac{x-y}{2}}|} \cdot \frac{1}{y}$

в) $\frac{\sqrt[3]{x} + \sqrt[5]{y} + \sqrt[7]{z}}{|x - y - z|}$

г) $\frac{-b - \sqrt{|b^2 - c^2|}}{a^2 + b^2 + 1}$

Задача 2 (4 точки). Дадени са целите числа a , b и c .

а) Да се дефинира булев израз, който има стойност *true*, ако съществува такова разместване на числата, при което те образуват строго монотонно растяща редица и има стойност *false*, в противен случай.

б) Да се намери отрицанието на булевия израз, дефиниран в а).

Задача 3 (5 точки). Дадени са целите числа a , b , c и d . Да се напише програма, която извежда на екрана средно-аритметичното на нечетните числа сред дадените. Ако няма нечетни числа, на екрана да се изведе съобщението „няма нечетни числа”.

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

$$S = \frac{2}{3!.4} + \frac{4}{5!.6} + \dots + \frac{2n}{(2n+1)!. (2n+2)}$$

където n е дадено естествено число.

Задача 5 (6 точки). Дадена е редица от цели числа, която съдържа точно веднъж числото 0 и има поне 2 елемента преди 0. Да се дефинира програмен фрагмент, който проверява дали в частта на редицата, започваща от първия елемент до елемента, равен на 0 има два последователни елемента, които се делят на 5. В случай, че има, да се изведат поредните номера на числата на първата такава двойка елементи. При реализиране на търсенето редицата да се обходи еднократно като се използва операторът за цикъл *do-while*.

Задача 6 (6 точки). Даден е масив a от $2n$ цели числа ($n > 0$ е дадено цяло число). Да се напише функция, която връща *true*, ако всяко число от втората половина на a е по-малко от всяко число от първата половина на a . За целта да се формулира и релизира подходяща задача за съществуване. Да не се влагат *return* и *break* в операторите за цикъл.

Задача 7 (4 точки). Какъв е резултатът от изпълнението на програмата?

<pre>#include <iostream> using namespace std;</pre>	
---	--

<pre> void f(int x, int* y, int& z) { int p = 1, q = 2; p = (q + z)/3; z = x + p; y = &p; cout << "f: x = " << x << endl; cout << "f: *y = " << *y<< endl; cout << "f: z = " << z << endl; cout << "f: p = " << p << endl; cout << "f: q = " << q << endl; return; } int main() { int p = 6, &q = p; f(p+q, &p, p); cout << "main: p = " << p << endl; cout << "main: q = " << q << endl; return 0; } </pre>	
---	--

Задача 8 (4 точки). Дадени са две редици от реални числа: a с n елемента и сортирана във възходящ ред и b с m елемента и сортирана в низходящ ред. Да се дефинира функция, която слива редиците a и b в нова редица c с $n+m$ елемента и сортирана в низходящ ред. За целта да се модифицира изветният алгоритъм за сливане *merge*.

Задача 9 (5 точки). Дадена е квадратна матрица a от порядък n от цели числа. Да се напише булева функция, която връща *true*, ако в a има стълб, всички елементи на който са удвоени нечетни числа.

Задача 10 (6 точки). Дадена е редица от числа a . Да се дефинира функция от по-висок ред, която прилага реалната едноаргументна числова функция f над елементите на a . Обхождането на редицата a е от началото към края й.

Задача 11 (5 точки). Дадени са първият член и частното на геометрична прогресия. Да се напишат рекурсивни функции за намиране на:

- а) n -тия член на прогресията;
- б) сумата от първите n члена на прогресията.

Задача 12 (6 точки). Даден е масив a от $2n$ цели числа ($n > 0$ е дадено цяло число). Да се напише рекурсивна функция, която връща *true*, ако всяко число от втората половина на a е по-малко от всяко число в първата половина на a .

Задача 13 (8 точки). Да се дефинира рекурсивна функция, която въвежда от клавиатурата, записан без грешка, израз от вида:

$$\langle \text{израз} \rangle ::= f(\langle \text{израз} \rangle, \langle \text{израз} \rangle, \langle \text{израз} \rangle) \mid g(\langle \text{израз} \rangle, \langle \text{израз} \rangle, \langle \text{израз} \rangle) \mid \langle \text{цифра} \rangle$$

където f и g са цели функции, дефинирани по следния начин:

$$f(x, y, z) = (x+y+z)/3,$$
$$g(x, y, z) = (|x-y| + |y-z| + |z-x|) \% 10.$$

Функцията да намира стойността на израза.