

ЦИКЛИЧЕСКИЕ ВЫЧИСЛИТЕЛЬНЫЕ ПРОЦЕССЫ

Цель работы: изучить приемы составления циклических алгоритмов, правила использования операторов for, while, do – while, а также операторов перехода continue и break.

Примеры решений

1. Введите два числа. Посчитать сумму чисел между ними.

```
#include <stdio.h>
void main(void)
{
    //определение переменных
    int i, a, b, sum=0;
    //ввод данных
    printf("Vvedite 2 chisla: ");
    scanf("%d%d", &a, &b);

    if(a > b)
    {
        i = a;
        a = b;
        b = i;
    }

    printf("%d", a);
    sum = a;
    for(i = a+1; i <= b; i++)
    {
        printf("%+d", i);
        sum += i;
    }

    printf("=%d\n", sum);
}
```

Разберем данный код подробно:

int i, a, b, sum=0; Определение переменных: **i**– циклическая переменная, **a** и **b** – переменные, куда вводятся значения с клавиатуры, а в переменной **sum** будет храниться сумма.

Цикл **for(i = a; i <= b; i++)** сработает только в том случае, когда **a < b**, поэтому до цикла написан условный оператор **if**, который меняет значение переменных **a** и **b**, если **a > b**.

printf("%+d", i); выводит на экран результатов число **i** со знаком.

2. Введите число, которое записывается в переменную n и является степенью. Посчитать 2^n .

```
#include <stdio.h>
void main(void)
{
    //определение переменных
    int n, st = 1, i;

    printf("Введите степень:");
    scanf("%d", &n);

    for(i = 1; i <= n; i++)
        st *= 2;
    printf("2^%d=%d\n", n, st);
}
```

В переменную **n** мы вводим степень двойки. Переменная **i** для прохода по циклу, а в переменную **st** мы записали сразу же 1, т.к. в ней мы будем считать степень двойки.

Предположим, в переменную **n**, мы ввели 3, тогда в переменной **st** должно храниться $2^3=8$.

Посмотрим работу цикла **for**, при **i=1** в переменной **st=st*2**, т.е. берется начальное значение переменной **st**, т.е. 1 умножается на 2, получаем 2 и снова записываем в переменную **st**, т.е. при **i=1** в переменной **st=2**.

При **i=2** в переменной **st=st*2**, т.е. берется значение переменной **st**, т.е. 2 умножается на 2, получаем 4 и снова записываем в переменную **st**, т.е. при **i=2** в переменной **st=4**.

При $i=3$ в переменной $st=st*2$, т.е. берется значение переменной st , т.е. 4, умножается на 2, получаем 8 и снова записываем в переменную st , т.е. при $i=3$ в переменной $st=8$.

При $i=4$ прекращается работа цикла. На экране результатов выводится $2^3=8$.

3. Ввести целое число и определить является ли данное число простым или нет.

```
#include <stdio.h>
```

```
void main(void)
{
    int n, i, f=1;

    printf("Vvedite chislo: ");
    scanf("%d", &n);

    for(i = 2; i <= n/2; i++)
    {
        if(!(n%i))
            { f = 0; break; }
    }

    if(f ==1)
        printf("\a\aChislo PROSTOE!!!!\n");
    else
        printf("NET!!!!\n");
}
```

4. Посчитать среднее арифметическое всех вводимых чисел, ввод чисел завершается, когда вводим 99.

4.1 используя оператора while

```
#include <stdio.h>
```

```
void main(void)
{
    int x , kol = 0;
    double sr = 0;

    printf("Vvedite chislo: ");
    scanf("%d", &x);

    while(x!=99)
    {
        kol++;
        sr += x;
        printf("Vvedite chislo:");
        scanf("%d", &x);
    }

    if ( kol ) sr /= kol;
    printf("srednee = %.2lf\n",sr);
}
```

4.2 используя оператора do/while

```
#include <stdio.h>
```

```
void main(void)
{
    int x , kol = 0;
    double sr = 0;

    do{
        printf("Vvedite chislo: ");
        scanf("%d", &x);

        kol++;
        sr += x;
    }while(x != 99);
}
```

```

    if ( kol )    sr /= kol;

    printf("srednee = %.2lf\n", sr);
}

```

Указать все преимущества и недостатки решения данной с помощью цикла *wile* и *do/while*.

5. Ввести число и распечатать те цифры данного числа, которые делятся без остатка на три (используя / и %).

```

#include <stdio.h>
void main()
{
    int n, z;
    printf("n= ");
    scanf("%d", &n);
    while( n )
    {
        z = n % 10; // последняя цифра числа n

        if (z % 3 == 0)
            printf("%d ", z);

        n = n / 10; //избавление от последней цифры числа n
    }
}

```

6. Вычислить $S = \sum_{k=1}^N \frac{1}{k}$, на печать выводить промежуточные и окончательный результаты.

```

#include <stdio.h>
void main(void)
{
    float s;
    int k,N;
    puts(" Введите N");
    scanf("%d", &N);

    for (s=0, k=1; k <= N; k++) {
        //В заголовке цикла можно выполнять и двойное
        //присваивание
        s += 1.0/k;
        printf(" \n k = %d  s = %f ", k, s);
    }

    printf("\n ОТВЕТ:  s = %f ", s);
}

```

Задачи для аудиторной и самостоятельной работы

Задача 1. Ввести целое неотрицательное число. Посчитать факториал данного числа.

Задача 2. Введите два числа. Первое число x – основание, второе число – степень. Посчитать x^n .

Задача 3. Ввести целое неотрицательное число N . Используя один цикл, найти сумму $1! + 2! + \dots + N!$ (выражение $N!$ – это N факториал). Чтобы избежать целочисленного переполнения, проводить вычисления с помощью вещественных переменных.

Задача 4. Введите целые положительные числа A и B ($A < B$). Вывести все целые числа от A до B включительно, при этом каждое число должно выводиться столько раз, каково его значение (например, число 3 выводится 3 раза).

Задача 5. Ввести целое число и определить, является ли данное число совершенным или нет. Целое число является совершенным, если его сомножители, включая 1 (но не само число), в сумме дают это число. Например, 6 – это совершенное число, т.к. $6=1+2+3$.

Задача 6. Вывести все простые числа в интервале от 2 до 10000.

Задача 7. Посчитать: количество нечетных, отрицательных чисел и произведение всех положительных чисел, ввод чисел завершается, когда вводим 0.

Задача 8. Вводим числа, ввод чисел заканчивается 0. Посчитать, сколько раз вводилось каждое из следующих чисел: -10, 5, 25 и 100.

Задача 9. Программа, которая находит наименьшее из нескольких вещественных чисел. Предполагается, что первое введенное число задает количество последующих вводимых чисел.

Задача 10. Ввести целое число и посчитать сумму и произведение цифр данного числа.

Задача 11. Вводим число, затем вводим символ ('+', '-', '*', '/', '='). Ввод завершается при вводе символа '=' Программа считает и выводит результат введенного выражения.

Задача 12. Дано целое число $N(>1)$. Вывести наименьшее из целых чисел K , для которых сумма $1+2+\dots+K$ будет больше или равна N , и саму эту сумму.

Задача 13. Разработать программу, которая должна определять заработную плату для каждого работника. Компания выплачивает каждому служащему почасовую зарплату за первые 40 часов работы и выплачивает в полуторном размере за все рабочие часы сверх 40. Дан список сотрудников, количество часов, отработанных ими, и почасовая ставка каждого сотрудника. Программа должна ввести эти данные для каждого сотрудника, распечатать и вывести на экран его суммарную зарплату.