

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Ильшат Ринатович Мухаметзянов

Должность: директор

Дата подписания: 13.07.2023 12:35:18

Уникальный идентификатор документа: aba80b84033c9ef196388e9ea0434f90a83a40954ba270e84bcb664f02d1d8d0

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Казанский национальный исследовательский

технический
университет им. А.Н. Туполева-КАИ»
(КНИТУ-КАИ)
Чистопольский филиал «Восток»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ
по дисциплине
ОСНОВЫ ПРОГРАММИРОВАНИЯ

Индекс по учебному плану: **Б1.О.12.01**

Направление подготовки: **09.03.01 Информатика и вычислительная техника**

Квалификация: **Бакалавр**

Профиль подготовки: **Вычислительные машины, комплексы, системы и сети**

Типы задач профессиональной деятельности: **проектный, производственно-технологический**

Рекомендовано УМК ЧФ КНИТУ-КАИ

Чистополь
2023 г.

Практическое занятие 1

Программирование простейших циклов

1. Вывести 10 раз текст с новой строки

Чистополь

```
printf (“\n Чистополь ”);
```

Операторы цикла:

Цикл с предусловием

```
i=0;
```

```
while (i<10)           пока условие истинно
```

```
{
```

```
printf (“\n Чистополь”);    вывод строки
```

```
i++;                       оператор
```

```
}
```

Цикл с постусловием

```
i=0;
```

```
do
```

```
{ printf (“\n Чистополь”);    вывод строки
```

```
i++;                       делать оператор
```

```
} while (i<10);           пока i <10
```

Цикл с параметром

```
for( i=0; i<10; i++)       пока параметр <10
```

```
printf (“\n Чистополь”);    вывод строки
```

2. Вычислить $1+2+3+ \dots + n$

3. $1 \cdot 2 \cdot 3 \cdot \dots \cdot N$

4. x^n

5. $S= 1+x+x^2/2! + \dots + x^n/n!$

6. Дано натуральное число n . Определить сумму цифр этого числа и количество цифр. Определить есть ли заданная цифра.

7. Вычислить $t + 2t + 3t + \dots + nt$

Практическое занятие 2

Условный оператор

1. Вычислить $y=1/x$. Учесть случай $x=0$. Оператор ветвления (оба var).
3 варианта if : if (x!=0) {y=1/x; printf();}else printf(); if(x==0); if(!x); (с else)
или просто

```
if (x!=0) printf (“\ny=%f”,1/x);
```

2. Записать следующие фрагменты программы без помощи оператора for. Дополнить их определением типов переменных.

```
for(r=-1,n=500; n>0; n--)
```

```
{ scanf("%f",&z);
```

```
if (z<r) r=z;
```

```
}
```

3. Записать следующие фрагменты программы с помощью оператора for. Дополнить их определением типов переменных.

```
y=0; m=100;
```

```
while (m >= 1)
```

```
{ scanf("%f",&x);
```

```
if(x<0) y=y-x;
```

```
else y=y+x;
```

```
m=m-1;
```

```
}
```

Задача 1. Составить программу построения (вывода) арифметической прогрессии для заданных начального члена, разности прогрессии, требуемого количества членов. Последовательность вида

t t+r t+2r t+3r ...

называют арифметической прогрессией, число t - начальным членом, а число r - разностью прогрессии.

Решение

```
/*    Программа 2.1. Арифметическая прогрессия
```

```
*/
```

```

#include <stdio.h>

main()
{
    int t;                /* начальный член          */
    int r;                /* разность прогрессии   */
    int kol;              /* количество членов     */
    scanf ("%d%d%d", &t, &r, &kol);
    printf ("\n%d", t);   /* вывод первого члена   */
    for (kol--; kol>0; kol--)
    { t += r;
      printf (" %d", t);  /* вывод очередного члена */
    }
}

```

Задания

1. Дано n – натуральное. Определить совпадают ли 1 и последняя цифры.

2. Дано n – натуральное. Определить четная ли разность 1 и последней цифры.

$$3. t \ t*r \ t*r^2 \ t*r^3 \ \dots \ t*r^n$$

Выполнить трассировку, составить схемы.

4. Записать следующие фрагменты программы без помощи оператора for. Дополнить их определением типов переменных.

a)	б)
for(j=0;j<10;j++)	for(r=-1, n=500; n>0; n--)
m[j] = 0;	{ scanf ("%f",&z);
if (z<r) r=z;}	

5. Записать следующие фрагменты программы с помощью оператора for. Дополнить их определением типов переменных.

a)	б)	в)
----	----	----

```

j=0;          y=0; m=100;          j=0; k=0;
while (j < 80)      while (m >= 1)      while (t[k]!='.')
{ s[j] = ' ';      { scanf("%f", &x);      { if (t[k]!='*')
  j++;              if(x<0) y = y - x;      { r[j]=t[k]; j++; }
}                  else    y = y+x;      k++;
  m = m-1;          }
}

```

6. Составить программу вычисления произведения двух натуральных чисел А и В, не используя операцию умножения.

Практическое занятие 3

Последовательная обработка символов

Задача 1. Дана последовательность символов, продолжающаяся до конца файла (при вводе с клавиатуры она заканчивается комбинацией клавиш Ctrl+Z). Выяснить, верно ли, что среди символов имеются все буквы слова "КНИТУ".

Решение.

```
/* Программа 1. Буквы слова " КНИТУ " */
#include <stdio.h>
main()
{ int i;          /* счетчик */
  char c;        /*текущий символ */
  int prov;      /* результат проверки (1 или 0) */
  char s[]="КНИТУ ";
  int priz[5]={0,0,0,0,0};/* признаки проверяемых букв */
  printf ("\nВведите последовательность символов\n");
  printf ("Завершение последовательности CTRL+Z\n");
  while ((c=getchar())!=EOF)
  { i=0; prov=1;
    while(i<5 && prov)
    { if (c==s[i]) { priz[i]=1; prov=0;}
      i++;
    }
  }
  prov = 1;
  for (i=0; i<5; i++)
  if (priz[i]==0) prov=0;
  if (prov) printf ("\n Есть все символы слова КНИТУ \n");
  else
  { printf("\n Нет символов ");
```

```
for (i=0; i<5; i++)
    if (!priz[i]) { putchar(s[i]); putchar(' '); }
}
```

Задачи

1. Дан текст произвольной длины, продолжающийся до конца файла.

Составить программы для решения следующих задач:

- а) Найти порядковый номер первой запятой.
 - б) Найти порядковый номер последней запятой.
 - в) длину каждой последовательности пробелов.
 - г) длину максимальной последовательности пробелов.
 - д) количество сочетаний АВ
 - е) количество слов с буквой 'а'
2. Определить длину каждого слова
 3. Определить максимальное слово, состоящее из цифр

Практическое занятие 4

Тема: Обработка массивов

Задача 1. Дано натуральное число $N > 0$ и последовательность из N действительных чисел. Напечатать сначала все отрицательные, а затем все остальные числа.

Решение

```
/*Программа 6.1. Печать отрицательных, затем остальных чисел */
#include <stdio.h>
#define N 100
main()
{ float a;          /* переменная для вводимых чисел      */
  float mas[N];     /* массив для неотрицательных чисел      */
  int i;           /* счетчик чисел                          */
  int n;           /* количество чисел                       */
  int k=0;         /* количество неотрицательных чисел     */
  int pr=1;        /* 1 - выводить заголовок, 0 - нет       */
  printf ("\nВведите количество чисел (не более 100)\n");
  scanf ("%d", &n);
  printf ("Введите последовательность из %d чисел\n",n);
  for (i=0; i<n; i++)
  { scanf("%f",&a);
    if (a<0)
    { if (pr) { printf ("\nОтрицательные числа:\n"); pr=0; }
      printf("%5.2f ",a);
    } else { mas[k]=a; k++; }
  }
  if (k)
  { printf ("\nНеотрицательные числа:\n");
    for ( i=0; i<k; i++) printf ("%5.2f ", mas[i]);
    putchar('\n');
```

```
    } else printf("\nНеотрицательных чисел нет.\n");  
}
```

Результаты работы программы

Тест 1

Введите количество чисел

10

Введите последовательность из 10 чисел

-1 4 3.2 7 -5.1 6 0 8.3 -9 11

Отрицательные числа:

-1.00 -5.10 -9.00

Неотрицательные числа:

4.00 3.20 7.00 6.00 0.00 8.30 11.00

Тест 2

Введите количество чисел

6

Введите последовательность из 6 чисел

-2.0 -8.1 -4.5 -10.9 -1.5 -5.1

Отрицательные числа:

-2.00 -8.10 -4.50 -10.90 -1.50 -5.10

Неотрицательных чисел нет.

Задача 2. Дан массив A из M действительных чисел. Известно, что $A[1]>0$ и что среди остальных элементов есть хотя бы одно отрицательное число. Найти сумму элементов, предшествующих первому отрицательному элементу.

Решение

```
/* Программа 2. Сумма элементов до отрицательного числа */  
#include <stdio.h>  
#define M 100  
main()  
{ float a[M]; /* массив действительных чисел */
```

```

float s=0;      /* сумма элементов до 1-го отрицательного */
int i;         /* счетчик */
int n;        /* количество элементов в массиве */
printf ("\nВведите количество элементов в массиве\n");
scanf ("%d",&n);
printf ("Введите массив из %d чисел\n",n);
for (i=0; i<n; i++)  scanf("%f",&a[i]);
if (a[0]<0) printf ("\nМассив начинается с отрицательного числа\n");
else
/* цикл суммирования положительных элементов до */
/* отрицательного или до конца массива */
{ i=0;
  do
    s += a[i++];      /* соответствует s = s + a[i]; i = i + 1; */
  while (a[i] > 0 && i < n);
  if (i==n)
  { printf ("\nВ массиве только положительные числа");
    printf ("\nСумма чисел = %5.2f\n",s);
  }
  else printf ("\nСумма до отрицательного = %5.2f\n",s);
}
}

```

Результаты работы программы

Тест 1

Введите количество элементов в массиве

6

Введите массив из 6 чисел

1 2 3 -4 5 6

Сумма чисел до отрицательного = 6.00

Тест 2

Введите количество элементов в массиве

6

Введите массив из 6 чисел

1 2 3 4 5 6

В массиве только положительные числа

Сумма чисел = 21.00

Тест 3

Введите количество элементов в массиве

6

Введите массив из 6 чисел

-1 2 3 4 5 6

Массив начинается с отрицательного числа

Задача 3. Целочисленная квадратная матрица размера $n \times n$ ($n \leq 50$) вводится по строкам. Составить программу определения номера столбца, имеющего максимальную сумму элементов, расположенных выше главной диагонали.

Решение.

Тест

Входная матрица:

Результат:

(точками отмечены элементы, не участвующие в задаче)

Номер столбца с максимальной суммой равен 3

$n=5$

	0	1	2	3	4
0	.	6	1	3	-2
1	.	.	4	-1	3
2	.	.	.	5	1
3	2
4

```

/*Поиск столбца с наибольшей суммой выше главной диагонали */
#include <stdio.h>
#define NMAX 50      /* Максимальный размер матрицы */
void main (void)
{ int m[NMAX][NMAX]; /* Матрица */
  int n;             /* Количество строк и столбцов */
  int i;             /* Индекс текущей строки */
  int j;             /* Индекс текущего столбца */
  int jmax;          /* Индекс столбца с максим. суммой */
  int s, smax;       /* Сумма текущего столбца и максим. */

  /* 1. Ввод матрицы m по строкам */
  scanf ("%d", &n);
  for (i=0; i<n; ++i)
    /* Ввод i-й строки */
    for (j=0; j<n; ++j) scanf("%d", &m[i][j]);
  /* 2.Поиск столбца с максим.суммой выше главной диагонали */
  jmax = 1;  smax = m[0][1];
  for (j=2; j<n; j++) /* Перебор столбцов */
  { /* Вычисление суммы j-го столбца */
    for (s=0, i=0; i<j; i++)
      s = s + m[i][j];
    if (s > smax) { smax=s; jmax=j; }
  }
  printf ("\nНомер столбца с максим-й суммой = %d", jmax);
}

```

Задача 4. Составить программу для решения следующей задачи. Заполнить квадратную матрицу $A = \{a_{i,j}\}$ ($i, j = 1, \dots, n$) значениями, вычисленными по формуле $a_{i,j} = i * j \% 5 + i - j$. Из матрицы получить вектор $X = \{x_i\}$ ($i = 1, \dots, n$). Элемент x_i вычислять как скалярное произведение i -ой

строки матрицы на столбец, содержащий первый по порядку максимальный элемент i -ой строки. Полученный вектор упорядочить по возрастанию методом последовательного нахождения минимума.

Решение

```
/* Программа 4. Обработка матрицы и вектора */
#include <stdio.h>
#define N 10 /*максимальный размер матрицы */
main()
{ int n; /* размер матрицы */
  float a[N][N]; /* матрица */
  float x[N]; /* вектор */
  int i,j,k; /* текущие индексы строк и столбцов */
  float m;
  /* Ввод матрицы */
  printf ("\nВведите размер массива\n");
  scanf ("%d",&n);
  for (i=0; i< n; i++)
    for (j=0; j< n; j++)
      a[i][j] = i * j % 5 + i - j;
  /* Печать матрицы */
  printf ("\nПолученная матрица\n");
  for (i=0; i< n; i++)
  { putchar('\n');
    for (j=0; j< n; j++) printf("%3.0f ",a[i][j]);
  }
  putchar('\n');
  /* Получение вектора из матрицы */
  /* Определение номера столбца, содержащего */
  /* первый по порядку максимальный элемент */
```

```

for (i=0; i<n; i++)
{
    k=0;
    for (j=1; j<n; j++)
        if (a[i][j] > a[i][k]) k = j;
    /* вычисление скалярного произведения */
    /* i-й строки на k-й столбец */
    x[i] = 0;
    for (j=0; j<n; j++)
        x[i] = x[i] + a[i][j] * a[j][k];
}
/* Печать вектора */
printf ("\nПолученный из матрицы вектор\n");
for (i=0; i<n; i++)
    printf ("%3.0f ", x[i]);
putchar('\n');
/* Упорядочивание вектора */
for (i=0; i<n; i++)
{
    j = i; m = x[i];
    for (k=i+1; k<n; k++)
        if (x[k] < m) { j = k; m = x[k]; }
    x[j]=x[i]; x[i]=m;
}
/* Печать упорядоченного вектора */
printf ("\nУпорядоченный вектор\n");
for (i=0; i<n; i++)
    printf ("%3.0f ",x[i]);
putchar('\n');
}

```

Результаты работы программы

Введите размер массива

6

Полученная матрица

0	-1	-2	-3	-4	-5
1	1	1	1	1	-4
2	3	4	0	1	-3
3	5	2	4	1	-2
4	7	5	3	1	-1
5	4	3	2	1	0

Полученный из матрицы вектор

-55 -10 11 27 36 20

Упорядоченный вектор

-55 -10 11 20 27 36

Задачи 5: Дана матрица действительных чисел A размера $n*n$ ($n \leq 40$).

Составить фрагменты программ для решения следующих задач.

Из матрицы A получить компоненты вектора $X[n]$ равные:

- а) суммам элементов строк;
- б) значениям средних арифметических элементов строк;
- в) числу отрицательных элементов в строке;
- г) сумме положительных элементов столбцов;
- д) количеству отрицательных элементов строк;
- е) наименьшим элементам строк;
- ж) наибольшим элементам столбцов;
- з) разностям наибольших и наименьших элементов строк;

В векторе X выбрать некоторое значение Z , равное соответственно:

- а) наибольшему элементу;
- б) номеру наименьшего элемента;
- в) сумме наибольшего и наименьшего элемента;
- г) наименьшему элементу;
- д) номеру наибольшего элемента;

- е) сумме элементов вектора;
- ж) среднему арифметическому элементов вектора;
- з) произведению элементов.

Задачи 6: Дана матрица действительных чисел размера $n*m$ ($n \leq 50$, $m \leq 70$). Составить фрагменты программ для решения следующих задач.

а) Вычислить сумму положительных элементов каждой строки и найти номер строки, в которой эта сумма наибольшая.

б) Найти минимальный элемент в столбце матрицы (из не равных нулю) и разделить на него все элементы этого столбца.

в) Найти наибольший положительный и наименьший отрицательный элементы.

г) В каждой строке переставить местами наибольший и наименьший элементы.

Задачи 7: Дана квадратная действительная матрица B размера $n*n$ ($n \leq 70$).

Составить фрагменты программ для решения следующих задач:

а) Вычислить сумму элементов, расположенных над главной диагональю.

б) Найти наибольший элемент главной диагонали и вывести всю строку, в которой он находится.

в) В каждом столбце матрицы найти максимальный элемент и обменять его с элементом, расположенным на главной диагонали.

г) Найти строку с наибольшим количеством отрицательных элементов и разделить элементы этой строки на минимальный элемент главной диагонали.

Задачи 8: Элемент матрицы назовем седловой точкой, если он одновременно является наименьшим в своем столбце и наибольшим в своей строке или наоборот. Найти индексы всех седловых точек заданной матрицы.

Задачи 9: Определить, является ли заданная целочисленная квадратная матрица магическим квадратом, т.е. такой, что суммы элементов во всех строках и столбцах одинаковы.

Задачи 10: Даны целые числа M и N и вещественная матрица из M строк и N столбцов, вводимая по строкам (строка за строкой). Составить программы решения следующих задач.

- а) Определить средние арифметические значения каждой строки.
- б) Определить средние арифметические значения каждого столбца.
- в) Индексы и значение максимального из элементов матрицы (одного из таких элементов).

Список литературы

Основная литература

1. Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке C++ : учебное пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев ; под ред. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2023. — 512 с. + Доп. материалы [Электронный ресурс]. — (Среднее профессиональное образование). - ISBN 978-5-8199-0699-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1916204> (дата обращения: 28.06.2023). — Режим доступа: по подписке.

2. Рейзлин, В. И. Язык C++ и программирование на нём : учебное пособие / В. И. Рейзлин. — 3-е изд., перераб. — Томск : ТПУ, 2021. — 206 с. — ISBN 978-5-4387-0975-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/246239> (дата обращения: 28.06.2023). — Режим доступа: для авториз. пользователей.

3. Ефимова Ю.В. Программирование на языке высокого уровня: Практикум. - Казань: Изд-во Казан. гос. техн. ун-та, 2012. - 32 с.

4. Чукич, И. Функциональное программирование на C++ : учебное пособие / И. Чукич ; перевод с английского В. Ю. Винника, А. Н. Киселева. — Москва : ДМК Пресс, 2020. — 360 с. — ISBN 978-5-97060-781-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/140597> (дата обращения: 28.06.2023). — Режим доступа: для авториз. пользователей.

5. Программирование на языке Си/А.В.Кузин, Е.В.Чумакова - М.: Форум, НИЦ ИНФРА-М, 2015. - 144 с. - Режим доступа: <http://znanium.com/bookread2.php?book=505194>

Дополнительная литература:

1. Язык Си: кратко и ясно: Учебное пособие / Д.В. Парфенов. - М.: Альфа-М: НИЦ ИНФРА-М, 2014. - 320 с. - Режим доступа: <https://znanium.com/read?id=356040>

2. Программирование графики на C++. Теория и примеры : учеб. пособие / В.И. Корнеев, Л.Г. Гагарина, М.В. Корнеева. — М. : ИД «ФОРУМ» : ИНФРА-М, 2017. — 517 с. - Режим доступа: <http://znanium.com/bookread2.php?book=562914>

3. Программирование на языке C++: Учебное пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев; Под ред. Л.Г. Гагариной. - М.: ИД ФОРУМ: ИНФРА-М, 2012. - 512 с. - Режим доступа: <http://znanium.com/bookread2.php?book=244875>