# Задание 2. Построение таблиц функций.

## Цель задания:

1. Получение навыков в использовании оператора цикла с параметром.
2. Знакомство с методами оптимизации программ.

## Постановка задачи:

Составить программу вычисления значений функции F(X) на отрезке [A,B] в точках X1=А+Н, где Н=(В-А)/М, М – заданное число.

## Содержание отчета:

1. Постановка задачи.
2. Текст программы.
3. Таблица результатов.
4. Блок-схема.
5. Таблица трассировки (остановиться после 5-ой точки).

## Методические указания:

1. Для задания значений Х и соответствующих значений функции следует использовать простые переменные.
2. Значение шага Н должно вычисляться один раз.
3. При изменении значения аргумента Х использовать оператор присваивания Х:=Х+Н, а не оператор с использованием операции умножения Х: = А + I•Н, что существенно сокращает время выполнения программы.

## Варианты задания

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер варианта | Функция | Параметры | | |
| А | В | М |
|  | sin(x2) | /6 | 2\*/3 | 10 |
|  | tg(x/2)+cos(x) | /2 |  | 20 |

# Задание 3. Организация циклов в программе

## Цель задания

1. Получение навыков в выборе и использовании операторов цикла.
2. Знакомство с итерационными процессами.

## Постановка задачи

Используя оператор цикла, найти сумму элементов, указанных в конкретном варианте. Результат напечатать, снабдив соответствующим заголовком.

## Содержание отчета

1. Постановка задачи.
2. Текст программы.
3. План отладки, тесты, результаты отладки на тестах.
4. Результат решения конкретного варианта.
5. Анализ допущенных ошибок.
6. Математические выкладки.
7. Блок-схема.
8. Таблица трассировки .

## Методические указания:

1. При определении суммы членов ряда следует использовать рекуррентную формулу для получения следующего члена ряда.

Например, требуется найти сумму ряда с точностью , общий член которого .

Для получения рекуррентной формулы вычислим отношение следующего члена ряда к текущему: , откуда .

***Рекомендация: лучше решать основную формулу в лоб*.**

1. Пример программы.

При составлении программы будем считать, что точность достигнута, если .

CONST 

VAR M: INTEGER; AN, SUMMA: REAL;

BEGIN

SUMMA: =0; M: =1; AN: =1/3,

WHILE AN> DO

BEGIN

SUMMA: = SUMMA + AN; N:=N+1;

AN: = AN\*(N+1)/2/(2\*N+1);

END;

WRITELN ( ‘Сумма=’, SUMMA,’ Последний член ряда=’, AN);

END.

Протокол работы программы:

Сумма = 4.727439Е – 01 Последний член ряда = 4.113534Е – 05

## Варианты задания

1. Найти сумму ряда с точностью , общий член которого 
2. Найти сумму целых положительных четных чисел, меньших 100.

# Задание 4. Работа с массивами чисел.

## Цель задания:

1. Получение практических навыков в работе с массивами.
2. Знакомство с алгоритмами упорядочения.

## Постановка задачи:

Для конкретного варианта составить массив исходных данных. Изучив алгоритмы упорядочения [[1]](#endnote-1), выбрать один из них. Написать программу, которая работает с любым допустимым набором данных. Входную информацию и результаты счета вывести на печать, снабдив их соответствующими заголовками.

## Содержание отчета:

1. Постановка задачи.
2. Текст программы.
3. Тесты. Результаты отладки на тестах.
4. Анализ допущенных ошибок.
5. Результаты счета.
6. Блок-схема.

## Методические указания:

1. Исходные данные должны включать и положительные числа, и отрицательные, и нули.
2. Пример программы. Дан двумерный массив различных вещественных чисел, содержащий 5 строк и 4 столбца. Строку, содержащую максимальный элемент массива, поменять местами со строкой, содержащей минимальный элемент:

## Варианты задания

1. Упорядочить по не убыванию массив, содержащий 15 вещественных чисел.

1. # ЗАДАНИЕ 5. РАБОТА С МНОЖЕСТВЕННЫМИ ТИПАМИ ДАННЫХ

   ## Цель задания

   1. Получение навыков в задании переменных множественного типа и выполнении простейших операций над ними.

   2. Знакомство с задачами, в которых целесообразно использовать переменные множественных типов.

   ## Постановка задачи

   Ознакомиться с конечным и упорядоченным множеством символов, определенным на используемой для выполнения задания ЭВМ.

   Составить программу для одного из вариантов.

   ## Содержание отчета

   1. Постановка задачи.

   2. Текст программы и результаты ее выполнения.

   3. Выводы.

   ## Методические указания

   Программа должна правильно работать для произвольного набора символов.**.** Входная строка символов может быть длиннее строки экрана терминала, при этом программа работает не с функцией EOLN, а с признаком конца строки, который задается программистом.

   ## Варианты задания

   1. цифры от '5' до '9' и знаки арифметических операций;

   # ЗАДАНИЕ 6. ОПЕРАЦИИ НАД МНОЖЕСТВАМИ

   ## Цель задания

   1. Получение навыков в организации ввода/вывода значений множественных типов.

   2. Получение практических навыков в выполнении операций над множествами.

   ## Постановка задачи

   Задан список объектов, включающий в зависимости от варианта названия ЭВМ или видов спорта. Известно, что в каждом техникуме имеется определенный набор вычислительных машин, а учащиеся каж­дой группы занимаются определенными видами спорта. Необходимо задать конкретные наборы ЭВМ (перечни видов спорта) для каждого техникума (каждой группы). Количество техникумов (групп) указано в варианте.

   Введя исходные данные, необходимо построить и распечатать множество, удовлетворяющее указанному и варианте условию.

   ## Содержание отчета

   1. Постановка задачи для конкретного варианта.

   2. Инструкция пользования программой.

   3. Текст программы и результаты ее выполнения.

   4. Выводы.

   ## Методические указания

   При выполнении задания следует учесть приемы программирования, использованные в приведеннойниже программе AMAF.

   Известен набор продуктов - хлеб, масло, сыр,молоко, имеющихся в ассортименте магазинов. В тримагазина доставлены отдельные виды этих продуктов. Требуется построить множества А, В,С**,** которыесодержат соответственно:

   - продукты, имеющиеся одновременно во всехмагазинах;

   - продукты, имеющиеся по крайней мере в одном из магазинов;

   - продукты, которых нет ни в одном из магазинов.

   program amaf;

   const n=3;

   type продукты=(хлеб, масло, молоко, сыр);

   ассорт=set of продукты;

   магазин=array[1..n] of ассорт;

   var m1:магазин;

   x:продукт;

   A,B,C,xm1:ассорт;

   i,j,iw:integer;

   begin

   for i:=1 to n do begin

   m1:=[];

   writeln('введите номера продуктов i‑ого магазина');

   repeat read(iw)

   case iw of

   1: x:=хлеб;

   2: x:=масло;

   3: x:=молоко;

   4: x:=сыр;

   end;

   xm1:=xm1+[x];

   until eoln;

   m1[i]:=xm1;end;

   A:=m1[1]; B:=[]; C:=[хлеб..сыр];

   for i:=1 to n do begin

   B:=B+m1[i]; (\*во всех\*)

   A:=A\*m1[i]; (\*по крайней мере в одном\*)

   C:=C‑B; (\*нет ни в одном\*)

   end;

   for i:=1 to do begin

   case i of

   1: writeln('продукты имеются одновременно во всех магазинах');

   2: writeln('ассортимент продуктов');

   3: writeln('продукты,которых нет ни в одном магазине');

   end;

   for x:хлеб to сыр do

   if x in A then

   case x of

   хлеб: writeln('хлеб');

   масло: writeln('масло');

   молоко: writeln('молоко');

   сыр: writeln('сыр'); end;

   if i:=1 then A:=B else A:=C; end;

   end.

   В программе АМАР в разделе типов задается список объектов (продуктов), определяющий базовый тип ПРОДУКТ, на котором определен множественный тип АССОРТ. Информация о наличии продуктов в n магазинах задается как массив множеств.

   В начале программы вложенные циклы обеспечивают ввод исходной информации. При этом во внутреннем цикле формируется множество ХМ1, характеризующее наличие товаров в одном магазине. По выходе из внутреннего цикла эта информация заносится и массив Ml. Затем строятся множества А, В, С с использованием операций над множествами (пересече­ния, объединения, вычитания). В конце программы вложенные цикла по I и Х обеспечивают распечатку полученных множеств. При этом используется оператор CASE.

   Протокол работы программы АМАF:

   ВВОДИ НОМЕРА ПРОДУКТОВ 1-ГО МАГАЗИНА =>1 3

   ВВОДИ НОМЕРА ПРОДУКТОВ 2-ГО МАГАЗИНА =>1 2 3

   ВВОДИ НОМЕРА ПРОДУКТОВ 3-ГО МАГАЗИНА =>1

   ПРОДУКТЫ, ИМЕЮЩИЕСЯ ОДНОВРЕМЕННО ВО ВСЕХ МагаЗИНАХ

   ХЛЕБ

   АССОРТИМЕНТ ПРОДУКТОВ

   ХЛЕБ МАСЛО СЫР

   ПРОДУКТЫ, КОТОРЫХ НЕТ НИ В ОДНОМ МАГАЗИНЕ

   МОЛОКО

   ## Варианты задания

   **1.** Задано множество вычислительных машин, которыми может быть обеспечен техникум: ЕС1035, ЕС1045, ЕС1036, ЕС1066, СМЗ, СМ4, ИСКРА. Известен набор машин, имеющихся в каждом техникуме. Количество техникумов (N) указано в варианте.

   . Требуется построить и распечатать множество, включающее в себя вычислительные машины:

   а) которыми обеспечены все техникумы (N=10);

   б) которые имеет хотя бы один техникум (N=10);

   в) которых нет ни в одном техникуме (М=10).

   Г) которые есть хотя бы в 3-х техникумах (N=10).

   **II.** В техникуме имеются секции по следующим видам спорта: лыжи, коньки, хоккей, футбол, регби и самбо. Известны виды спорта, которыми занимаются учащие­ся каждой группы. Количество групп N указано в ва­рианте.

   1. Требуется построить и распечатать множество видов спорта:

   а) которыми занимаются во всех группах (N=10);

   б) которыми занимаются хотя бы в одной группе (N=l6);

   в) которыми не занимаются ни в одной из групп (N=10).

   Г) которыми занимаются хотя бы в 5-и группах (N=10).

   # ЗАДАНИЕ 7. РАБОТА С КОМБИНИРОВАННЫМИ ТИПАМИ ДАННЫХ

   ## Цель задания

   1. Получение навыков в организации ввода/вывода значений комбинированных типов данных.

   2. Получение практических навыков программирования задач с использованием записей.

   ## Постановка задачи

   Составить список учебной группы, включающей 25 человек. Для каждого учащегося указать дату рождения, год поступления в техникум, курс, группу, оценки каждого года обучения. Информацию о каждом учащемся оформить в программе в виде записи. Совокуп­ность записей объединить в массив.

   Составить программу, которая обеспечивает ввод полученной информации, распечатку ее в виде таблицы, а также распечатку информации согласно конкретному варианту.

   ## Содержание отчета

   1. Постановка задачи для конкретного варианта.

   2. Исходная информация.

   3. Текст программы и результаты ее выполнения.

   1. Анализ допущенных ошибок.

   ## Методические указания

   1. Символьную информацию в программе следует задавать в виде упакованного массива.

   2. Пример программы.

   Программа ZAP вводит информацию о пяти студентах: фамилию, год рождения, год поступления в техникум и оценки последней сессии. Результатом ра­боты программы является список отличников. В программе показаны возможности организации работы с записями с использованием оператора присоединения:

   program zap;

   type студент=record

   фио: array[1..10] of char;

   годрож:1965..1975;

   годпост:1985..1995;

   оценки: (алг, ист, эвм):1..5;

   end;

   var группа:array[1..5] of студент;

   s1:студент;

   i,j:integer;

   begin

   with s1 do begin

   for i:=1 to 5 do begin

   for j:=1 to 10 do read(фио[j]);

   readln(годрож,годпост);

   with оценки do readln (алг, ист, эвм);

   группа[i]:=s1;

   end;

   end;

   writeln('список отличников');

   for i:=1 to 5 do with группа[i],оценки do

   if ((алг=5) and (ист=5) and (эвм=5)) then

   writeln(фио,годрож,годпост);

   end.

   ## Варианты задания

   1.Распечатать анкетные данные студентов, успевающих на 4 и 5.

   2.Распечатать список студентов, фамилии которых начинаются с буквы А, и их оценки за все время обучения.

   # ЗАДАНИЕ 8. РАБОТА С ВНЕШНИМИ ФАЙЛАМИ

   ## Цель задания

   1. Ознакомление с возможностями организации файлов на внешних носителях в используемой ЭВМ.

   2. Получение практических навыков работы с внешними файлами.

   ## Постановка задачи

   Подготовить данные об абитуриентах, поступающих в техникум. Информацию о каждом абитуриенте оформить в виде записи, содержащей следующие поля:

   1. Фамилия, имя, отчество.

   2. Год рождения.

   3. Год окончания школы.

   4. Оценки в аттестате.

   5. Признак - нуждается ли в общежитии.

   6. Оценки вступительных экзаменов.

   Разработать программу записи подготовленных данных во внешний файл и программу обработки созданного внешнего файла.

   I. Удалить из внешнего файла все записи, удовлетворяющие условию, заданному в варианте, и распечатать информацию, оставшуюся в файле.

   II. Добавить N записей в начало (конец) внешнего файла и распечатать записи полученного файла согласно конкретному варианту.

   ## Содержание отчета

   1. Постановка задачи.

   2. Анкетные данные абитуриентов.

   3. Тексты программ и протоколы их работы.

   4. Выводы.

   ## Методические указания

   1. При подготовке исходных данных необходимо учесть, что выходная информация программы обработки внешнего файла должна составлять не менее одной четверти от входной.

   Варианты задания:

   I. Из внешнего файла, содержащего исходные данные, удалить записи, соответствующие:

   1. Абитуриентам, получившим хотя бы одну оценку 2.

   2. Абитуриентам старше семнадцати лет, имеющим отличный аттестат.

   II.Распечатать полученный файл. П. Используя внешний файл, содержащий исходные данные, добавить N записей и распечатать:

   1. Список абитуриентов, имеющих в аттестате оценки только 5; N=2.

   2.Список абитуриентов, получивших на вступительных экзаменах одну оценку 3; N=4.

   # ЗАДАНИЕ 9. РАБОТА С УПОРЯДОЧЕННЫМИ ФАЙЛАМИ

   ## Цель задания

   1. Ознакомление с возможностью добавления информации в упорядоченный внешний файл.

   2. Получение навыков в организации работы с внешними файлами.

   ## Постановка задачи

   Подготовить список из N наименований товаров. Конкретное значение N указано в варианте. Информация о каждом товаре содержит:

   1. Название товара.

   2. Цену.

   3. Год выпуска.

   4. Количество.

   Список должен быть упорядочен по одному из параметров, указанных в конкретном варианте. Сведения по каждому товару необходимо представить в виде записи.

   Разработать программу, которая заносит во внешний файл записи упорядоченного списка, и программу, которая добавляет в сформированный внешний файл данные об М товарах, при этом, не нарушая упорядоченности исходного файла. Если среди добавляемых товаров встречается товар, сведения о котором в файле уже есть, то необходимо их обновить, т. е. старую запись исключить.

   ## Содержание отчета

   1. Постановка задачи.

   2. Упорядоченный исходный список товаров и перечень добавляемых товаров.

   3. Тексты программ и протоколы их работы.

   4. Анализ допущенных ошибок.

   ## Методические указания

   1. При подготовке исходных данных в качестве наименований товаров можно взять названия инструментов, книг, школьно-письменных принадлежностей и т. п. Для них соответственно указать цену, год выпуска, количество.

   ## Варианты задания

   Товары упорядочены по неубыванию года выпуска, N=9; M=4.

   Товары упорядочены по неубыванию цены, N=13, М=4. [↑](#endnote-ref-1)