**Дисциплина «Алгоритмы решения прикладных задач»**

**Рабочая тетрадь 5.**

**Рекурсия**

|  |  |
| --- | --- |
| **Теоретический материал** | |
| **Рекурсия** — это такой способ организации вспомогательного алгоритма (подпрограммы), при котором эта подпрограмма (процедура или функция) в ходе выполнения ее операторов обращается сама к себе. То есть в теле функции она вызывает саму себя.  Практически любую рекурсивную функцию можно переписать нерекурсивным образом с применением циклов и условных операторов.  Например, вычисление факториала числа *N*, т. е. вычисление произведения всех чисел от 1 до *N* (в математике обозначается *N*!) нерекурсивным способом можно на C++ записать в виде следующего цикла;  int N=10; //будем вычислять факториал числа 10  int fact = 1; //переменная для «накопления» значения факториала  for(int i=1; i<=N; i++)  {  fact\*=i;  }  //По окончании цикла в переменной fact будет факториал числа *N*  Рекурсивно посчитать факториал можно, написав функцию, которая при выполнении вызывает себя. Такое возможно, поскольку N!=N\*(N-1)!  int factorial(int i)  {  if (i==0) return 1;  else return i\*factorial(i-1);  }  *Пример 1 демонстрирует вычисление факториала рекурсивным и нерекурсивным образом.* | |
| **Пример 1** | |
| ***Задача:*** | |
|  | Написать на языке C++ программу, в которой факториал числа N считается с помощью цикла и с помощью рекурсивной процедуры. |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  |  |
| **Задание 1** | |
| ***Задача:*** | |
|  | Написать программу с рекурсивной функцией нахождения НОД двух чисел по алгоритму Евклида (*алгоритм объяснен в рабочей тетради 2, за основу можно взять нерекурсивный пример 1 из рабочей тетради 2*). Рекурсивно НОД двух чисел определяется следующим образом:    Выход из рекурсии происходит, когда a=b, это и есть НОД |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  |  |
| **Задание 2** | |
| ***Задача:*** | |
|  | Написать рекурсивную подпрограмму вычисления чисел Фибоначчи. Xn=Xn-1+Xn-2; X0=1; X1=1 (см. подсказки в слайдах с лекции) |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  |  |
| **Задание 3** | |
| ***Задача:*** | |
|  | Переписать программу из рабочей тетради 1 для нахождения корня уравнения методом половинного деления, реализовав метод через рекурсивную процедуру (вариант тот же, что и в рабочей тетради 1, см. подсказки в слайдах с лекции) |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  |  |
| **Задание 4\*** | |
| ***Задача:*** | |
|  | Написать программу с рекурсивной функцией для нахождения суммы цифр числа. |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  |  |
| **Задание 5** | |
| ***Задача:*** | |
|  | Написать программу с рекурсивной функцией для нахождения значения следующей функции |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  |  |
| **Задание 6\*** | |
| ***Задача:*** | |
|  | У нас есть список с числами и другими списками, в которых тоже есть  числа и списки:    Напишите на Python рекурсивную функцию, которая выводит все числа (только числа). |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  |  |
| **Задание 7\*** | |
| ***Задача:*** Есть последовательность так называемых треугольных чисел, которая начинается как 1, 3, 6, 10, 15, 21 и продолжается с N-го числа в шаблоне, равного N плюс предыдущее число. Например, седьмое число последовательности - 28, то есть 7 (номер числа N) плюс 21 (предыдущее число последовательности). Напишите рекурсивную функцию, которая принимает номер числа N и возвращает соответствующее число последовательности. Например, при передаче этой функции значения 7 она должна вернуть 28. | |
|  |  |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  |  |
| **Задание 8\*** | |
| ***Задача:***  Есть лестница из N ступенек, а человек может преодолеть одну, две или три ступеньки за раз. Сколько есть возможных «способов» подъема по такой лестнице? Напишите функцию, которая вычисляет число таких способов для N ступенек.    На рисунках приведены некоторые (но не все!) возможные варианты для лестницы из 5 ступенек | |
|  |  |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  |  |