Национальный исследовательский ядерный университет
«МИФИ» Институт интеллектуальных кибернетических
систем

Кафедра №12 «Компьютерные системы и технологии»

ОТЧЕТ

О выполнении лабораторной работы №1

«Алгоритмизация обработки целых чисел»

Студент: Титов Иван Андреевич.

Преподаватель: Уваров М.П.

Группа: Б23-901

Москва – г .2023

Лабораторная работа №1

Содержание:

I. Подготовительная часть (структуризация условий задачи)
II. Алгоритмическая часть
   1. Разработка блок-схемы
   2. Распределение типов данных для предполагаемых значений
   3. Разработка системы тестов (подготовка входных значений + системы считывания входных данных):
      А) На формат входных данных
      Б) На однозначность алгоритма и на адекватность выходных данных
III. Написание кода
IV. Проверка работоспособности программы и оценка эффективности программы
V. Выводы

I.Подготовительная часть

Лабораторная работа № 1 «Алгоритмизация обработки целых чисел»

Необходимо спроектировать и реализовать на языке С программу, осуществляющую, в соответствии с индивидуальным заданием, обработку одного или нескольких целых чисел.

Примечания:

1. Логически законченные части алгоритмов решения задачи должны быть оформлены в виде отдельных функций с параметрами. Использование глобальных переменных не допускается.
2. Программа должна содержать **не менее** двух функций.
3. Программа должны осуществлять проверку корректности вводимых данных и, в случае ошибок, выдавать соответствующие сообщения, после чего завершать работу.

Вариант №13

Задание

Из заданного диапазона целых чисел найти такие, квадрат которых равен кубу суммы их цифр.

Цель: создать программу, которая удовлетворяет требованиям.

Требования:

1. Принимать на вход граничные значения диапазона (задание диапазона), которые принадлежат к множеству целых чисел. Если входные данные не являются двумя целыми числами, требуется вывести ошибку и прервать выполнение программы
2. Оформлять логически законченные части алгоритма (и кода) в виде функций
3. Не использовать глобальные переменные
4. Алгоритм должен содержать 2 и более функции
5. Выводить целые числа из заданного диапазона, которые удовлетворяют условию

Примечания к решению:

1. Значения диапазона, выходящие за пределы ограничения для типа данных, будут приводить к выводу ошибки и прерыванию выполнения программы
2. Стоит учитывать, что существуют «небезопасные» операторы, которые могут приводить к неадекватному поведению программы (это характерно для случаев с неадекватными входными данными). Среди них:

1. Scanf ()/printf (): в силу необходимости ввода в качестве аргумента функции спецификатора преобразования, могут работать с ограниченным диапазоном входных значений. Но данные функции небезопасны из-за того, что в случае ввода значений, не соответствующих спецификатору, возникающие ошибки не фиксируется, значения на выходе становятся непредсказуемыми. Даже если проверять значение функции (к примеру, if (scanf(…) != 1 *Вывод ошибки*), то игнорируются слишком большие входные значения (они становятся непредсказуемыми из-за типа данных signed int, который при переполнении ведёт себя непредсказуемо)
2. Gets (): обеспечивает ввод в массив строки. Из-за необходимости объявления размера массива заранее возможна некорректная работа программы вследствие переполнения массива (в т.ч. системная ошибка)
   Поэтому не представляется возможным сделать в достаточной мере безопасным (а также «массовым») ввод и эксплуатацию данных с помощью одной лишь библиотеки <stdio.h>. В частности требуется работа со строками и/или массивами (fgets(), strchr()).

3. При вводе граничных точек они должны быть заведомо отсортированными, иначе будет выводиться ошибка и программа будет прервана

II.Алгоритмическая часть

Программа содержит следующие функции: main(), degofnum(), sumofnum() (в двух версиях: через остатки и через строку), mod(), inputproc().

Функции имеют свои специфический, закрепленные за ними операции (что обеспечивает функциям логическую завершенность):

1. Main(): объявление основных переменных: значения граничных точек (n1, n2) и сравниваемых значений для каждого обрабатываемого числа из диапазона (a, b)
2. Inputproc(): ввод значений, обработка значений на их адекватность (по трём критериям)[1][2]
3. Degofnum(): возведение числа в заданную степень
4. Sumofnum(): подсчёт суммы цифр числа
5. Mod(): нахождение модуля числа[3]

Поэтому требование №2 можно объективно считать выполненным.

Вызовы функций осуществляются в основном с помощью стандартного оператора return. Однако у функций main() и inputproc() сообщение иного рода: в main() создаются две переменных типа int (n1 и n2), которые в качестве указателей передаются при вызове функции inputproc() и уже в обработанном виде возвращаются (перезаписываются) в функции main().

Следующие переменные имеют тип данных unsigned int: n1, n2, a, b, i, j, z, nind, sm – по следующим причинам:

1. Int т.к. требуется совершать операции с целыми числами
2. Программа не предусмотрена для работы с большими числами, значит, нет необходимости в таких альтернативах с большей «вместимостью» (занимаемым местом в памяти), как double, long int и т.п.
3. Unsigned т.к. программа предусмотрена для работы с отрицательными граничными точками (и соответственно с отрицательными значениями диапазона)

Следующие переменные имеют тип данных char: strnum1, strnum2 – по следующей причине:

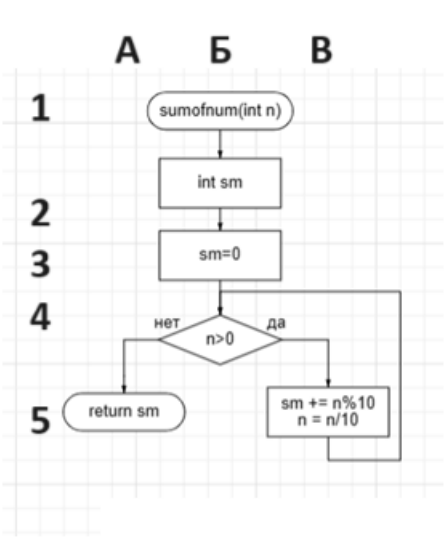
1. Для безопасной обработки данных (для того, чтобы ошибочные входные данные не игнорировались, а приводили к ошибке и

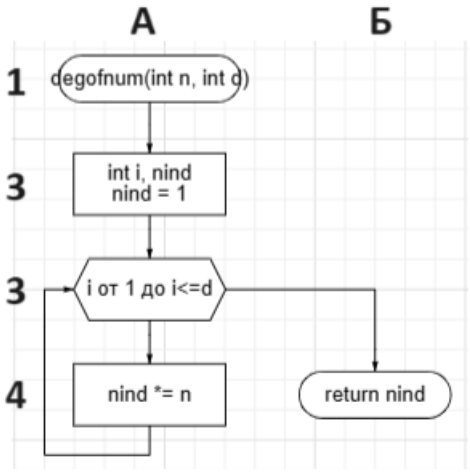
[1] Функция выводит ошибку только при вводе любых символов кроме чисел. Дробные числа преобразуются в целые путём отбрасывания дробной части (когда они присваиваются типу данных int). Поэтому справедливо отметить, что требование 1 выполняется лишь отчасти

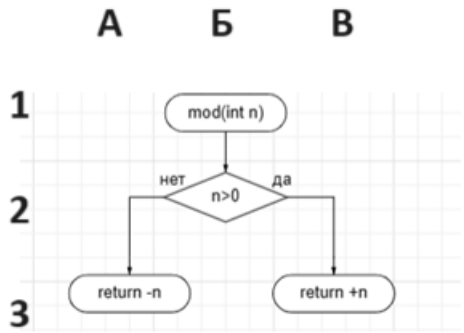
[2] Ограничение по величине значений, по сути, представляет из себя ограничение на длину вводимой строки (т.е. максимально допустимое отриц. число -9 999 999, положительное 99 999 999, с учётом того, что у строк существует занимающий одну ячейку конечный символ)

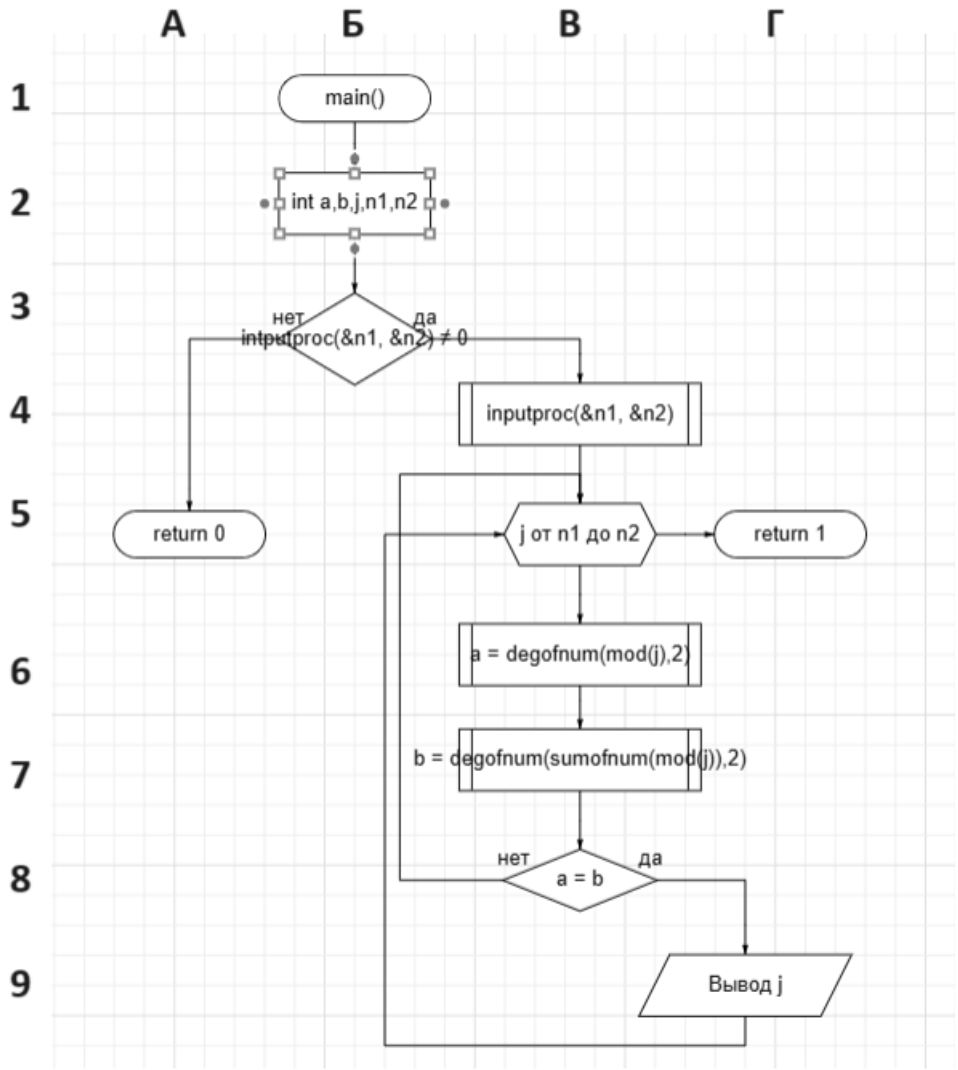
[3] Применение функции взятия модуля от числа не должно быть необходимым в программе. Но, учитывая возникающие трудности в работе программы без него и учитывая то, что от этого адекватность выходных значений не теряется, его применение необходимо в данном случае

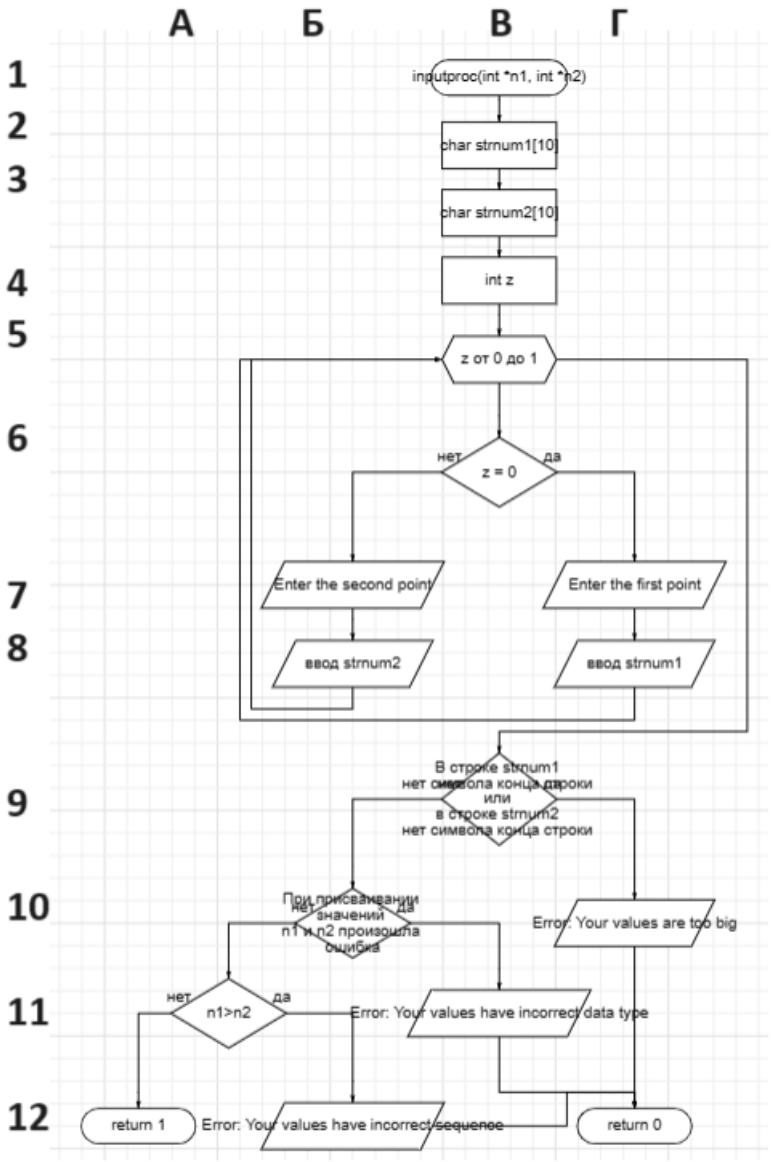
завершению программы) используется первичный ввод данных в качестве строки в буфере и дальнейшая их проверка








А
Б
В
Г
1
2
3
4
5
6
7
8
9
main()
int a,b,j,n1,n2
нет
да
inputproc(&n1, &n2) ≠ 0
inputproc(&n1, &n2)
return 0
j от n1 до n2
return 1
a = degofnum(mod(j),2)
b = degofnum(sumofnum(mod(j)),2)
нет
да
a = b
Вывод j


А
Б
В
Г
1
2
3
4
5
6
7
8
9
10
11
12
inputproc(int *n1, int *n2)
char strnum1[10]
char strnum2[10]
int z
z от 0 до 1
нет
z = 0
да
Enter the second point
Enter the first point
ввод strnum2
ввод strnum1
В строке strnum1
нет символа конца строки
или
в строке strnum2
нет символа конца строки
нет
При присваивании
значений
n1 и n2 произошла
ошибка
да
Error: Your values are too big
нет
n1>n2
да
Error: Your values have incorrect data type
return 1
Error: Your values have incorrect sequence
return 0

III.Код

```c
#include <stdio.h>
#include <string.h>
int degofnum(int n, int d) { //функция, считающая степень d числа n
	int i, nind;
	nind = 1;
	for (i = 1; i <= d; i++) {
		nind *= n;
	}
	return nind;
}
int mod(int n){
	if (n>0) return n;
	else return -n;
}
int inputproc(int *n1, int *n2){//проверка вводимых данных на: их тип, их упорядоченность, на их длину (что есть также ограничение на тип данных)
	char strnum1[10], strnum2[10];
	int z;
	for (z = 0; z < 2; z++) {
		if (z == 0) {
			printf("Enter the first point\n");
			fgets(strnum1, sizeof(strnum1), stdin);
		}
		else {
			printf("Enter the second point\n");
			fgets(strnum2, sizeof(strnum2), stdin);
		}
	}
	if (sscanf(strnum1, "%d", &*n1) != 1 || sscanf(strnum2, "%d", &*n2) != 1) {
		printf("Error: Your values have incorrect data type\n");
		return (0);
	}
	if (strchr(strnum1, '\n') == 0 || strchr(strnum2, '\n') == 0) {
		printf("Error: Your values are too big\n");
		return (0);
			}
	if (*n1>*n2){
		printf("Error: Your values have incorrect sequence\n");
		return (0);
	}
}
//int sumofnum(int n) { //функция, считающая сумму цифр числа с помощью строк
//	char k[20];
//	int sm, i;
//	sprintf(k, "%d", n); //перевод числа n в строку k
//	for (i = 0; i < strlen(k); i++) {  //пошаговое сложение к нулевой сумме цифр числа
//		sm += k[i] - '0';
//	}
//	return sm;
//}
int sumofnum(int n) { //функция, считающая сумму цифр
	int sm;
	sm = 0;
	while (n>0){
		sm += n%10;
		n = n/10;
	}
	return sm;
}
int main() {
```

```
        int n1,n2,j,a,b;
        if (inputproc(&n1,&n2)!=0){
                for (j = n1; j <= n2; j++) { //перебор чисел из диапазона
                        a = degofnum(mod(j), 2);
                        b = degofnum(sumofnum(mod(j)), 3);
                        //printf ("%d %d\n", a, b);
                        if (a == b) printf("%d\n", j);
                }
                        }
        else return (0);
        return (1);
}
```

IV.Проверка входных значений

Входные значения (1/2)	Ожидаемый вывод	Реальный вывод
10/Слово	Ошибка	Enter the first point 10 Enter the second point Слово Error: Your values have incorrect data ty
1 999 999 999 /2 999 999 999	Ошибка	Enter the first point 1999999999 Enter the second point Error: Your values are too big
100/10	Ошибка	Enter the first point 100 Enter the second point 10 Error: Your values have incorrect sequenc
0/1000	0, 1, 27	Enter the first point 0 Enter the second point 1000 0 1 27
-1000/1000	-27,-1,0,1,27	Enter the first point -1000 Enter the second point 1000 -27 -1 0 1 27

V.Выводы

В ходе работы над лабораторной работы №1 были освоены азы языка программирования C: типы данных, объявление/вызов функций, способы ввода/вывода значений, логические и арифметические операции с операндами.

Помимо того, немаловажным является приведение интуитивных представлений о построении алгоритмов и соответственно блок-схем к начальному (но, тем не менее, хоть сколько-нибудь структурированному) пониманию этих объектов.

Создание и оформление лабораторной работы – сам по себе процесс очень важный, т.к. это есть по сути первый проект подобного рода, в котором необходимо правильно изложить свои мысли и знания, представить их и впоследствии завершённый продукт.

Конечно, в желаниях было сделать куда более совершенный проект, но, в конце концов, реальность это всегда что-то среднее между «хочу» и «могу». Тем не менее, нереализованные желания побуждают учиться, думать и практиковаться.