|  |  |
| --- | --- |
|  | **Отчёт по лабораторной работе** № 10 по курсу 1 ­  студента группы M80-108Б-18 Коростелев Д.В. , № по списку 12  Адреса www, e-mail, jabber, skype  Работа выполнена: “ “ 200 г.  Преподаватель: Поповкин А.В. каф.806  Входной контроль знаний с оценкой  Отчёт сдан “ “ 20 г., итоговая оценка  Подпись преподавателя |

1. **Тема**: Техника работы с целыми числами. Системы счисления.
2. **Цель работы**: составить программу на языке Си в целом типе данных, которая для любых допустимых и корректно записанных чисел этого типа в десятичном изображении, поступающих на стандартный ввод программы, выполняет указанное вариантом действие над их значением.
3. **Задание** (*вариант №12* ): перенести ведущие нули десятичного представления в младшие разряды.
4. **Оборудование** (*лабораторное*):

ЭВМ Cameron, процессор Intel Core2 Duo CPU E8500 3.16ГГц, имя узла сети cameron с ОП 16029 МБ

НМД 50 ГБ. Терминал gnome адрес 172.16.80.213. Принтер лазерный принтер с технологией spooling Другие устройства

*Оборудование ПЭВМ студента, если использовалось:*

Процессор Intel Core i7-7700HQ 4x 2.808ГГц, ОП 3945 МБ, НМД 48,9 ГБ. Монитор

Другие устройства

1. **Программное обеспечение** (*лабораторное*):

Операционная система семейства UNIX , наименование UBUNTU версия 16.04

Интерпретатор команд bash версия 4.3.48

Система программирования gcc версия 7.3.0

Редактор текстов nano версия 2.9.3

Утилиты операционной системы gсс, nano

Прикладные системы и программы

Местонахождения и имена файлов программ и данных

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства UNIX , наименование UBUNTU версия 18.04

Интерпретатор команд bash версия 4.3.48

Система программирования gcc версия 5.4.0

Редактор текстов nano версия 2.5.3

Утилиты операционной системы gсс, nano

Прикладные системы и программы

Местонахождения и имена файлов программ и данных

1. **Идея, метод, алгоритм** решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальное описание с пред- и постусловиями)

Так как в задаче требуется перенести ведущие нули, а ведущие нули не сохраняются в целочисленных типах данных, будем сохранять каждую цифру числа в символьном виде. Обрабатывать данные будем от начала слова до пробела, определить является ли символ пробелом или переходом на следующую строку будет определять с помощью функции \_isspace. Если с начала слова идут 0, то считаем их кол-во, как только встречаем другое число, перестаем считать нули и последующих чисел начинаем формировать число (до этого переведя числа в целочисленный тип с помощью функции char\_to\_int) c помощью функции form\_number (в ее основе лежит функция a = a\*10 + b , где b - цифра считанная на данной итерации, а a – число, сформированное из прошлых цифр по той же формуле)

Когда встретим пробел, переносим ранее посчитанные нули в конец числа, умножая числа на 10 столько раз, сколько встретили 0 в начале числа. Повторяем пока не встретим символьный указатель \n.

После этого проверяем, встретилось ли какое-нибудь число до \n (это нужно, для того чтобы не потерять последнее число, находящееся вплотную с пробелом), если да до повторяем все действия с числом, описанные выше. Завершаем работу программы.

1. **Сценарий выполнения работы** [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты, либо соображения по тестированию].

**Тесты**

|  |  |
| --- | --- |
| **Исходные данные** | **Результат** |
| 01 | 10 |
| 0114 | 1140 |
| 0002 | 2000 |
| 0003 0004 | 3000 4000 |
| 003 567 909 | 300 567 909 |
| 003003 123 055604 | 300300 123 556040 |

*Пункты 1-7 отчёта составляются* ***строго до*** *начала лабораторной работы.*

Допущен к выполнению работы. Подпись преподавателя

1. **Распечатка протокола** (подклеить листинг окончательного варианта программы с текстовыми примерами, подписанный преподавателем)

|  |
| --- |
| dmitry@dmitry-VirtualBox:~/lubs/12$ cat 12prog.c  #include<stdio.h>  int char\_to\_int(char s) { //перевод чисел из char в int  if (s == '0')  return 0;  if (s == '1')  return 1;  if (s == '2')  return 2;  if (s == '3')  return 3;  if (s == '4')  return 4;  if (s == '5')  return 5;  if (s == '6')  return 6;  if (s == '7')  return 7;  if (s == '8')  return 8;  if (s == '9')  return 9;  }  int \_isspace(char s) { //функция, которая проверяет записан ли пробел  if (s == ' ' || s == '\n')  return 1;  else return 0;  }  int form\_number(int a, int b) { //функция, которая собирает число  a = a \* 10 + b;  return a;  }  int add\_zero(int a, int n) { //функция, которая добавляет нули к числу  for (int i = 0; i < n; i++)  a \*= 10;  return a;  }  int main(void) {  char symbol;  int b\_space, number\_of\_zero = 0, number\_1, number\_2 = 0, b\_start\_word = 1, length = 0, b\_was\_space = 1;  //b\_space – встретили пробел или нет, number\_of\_zero – количество нулей  //number\_1 – считанное число, number\_2 – формируемое число,  //b\_start\_word – начало слова или нет, length – длина слова,  //b\_was\_space – встречали ли пробел ранее  while((symbol = getchar())!='\n'){  b\_space = \_isspace(symbol);  if (symbol == '0' && b\_start\_word == 1)  {  number\_of\_zero++;  }  else if(symbol != ' ')  b\_start\_word = 0;  if (b\_space == 0) {  number\_1 = char\_to\_int(symbol);  number\_2 = form\_number(number\_2, number\_1);  length++;  b\_was\_space = 0;  }  else {  if (length > 0) {  number\_2 = add\_zero(number\_2,number\_of\_zero);  printf("%d ", number\_2);  length = 0;  number\_2 = 0;  b\_was\_space = 1;  b\_start\_word = 1;  number\_of\_zero = 0;  }  else if(b\_was\_space == 0)  {  printf(" ");  b\_was\_space = 1;  }  }  }  if (number\_2 != 0)  {  number\_2 = add\_zero(number\_2,number\_of\_zero);  printf("%d",number\_2);  }  printf("\n");  return 0;  }  dmitry@dmitry-VirtualBox:~/lubs/12$ gcc -std=c99 -ansi -pedantic -Wall -g -v -o 12.out  Using built-in specs.  COLLECT\_GCC=gcc  COLLECT\_LTO\_WRAPPER=/usr/lib/gcc/x86\_64-linux-gnu/7/lto-wrapper  OFFLOAD\_TARGET\_NAMES=nvptx-none  OFFLOAD\_TARGET\_DEFAULT=1  Target: x86\_64-linux-gnu  Configured with: ../src/configure -v --with-pkgversion='Ubuntu 7.3.0-27ubuntu1~18.04' --with-bugurl=file:///usr/share/doc/gcc-7/README.Bugs --enable-languages=c,ada,c++,go,brig,d,fortran,objc,obj-c++ --prefix=/usr --with-gcc-major-version-only --program-suffix=-7 --program-prefix=x86\_64-linux-gnu- --enable-shared --enable-linker-build-id --libexecdir=/usr/lib --without-included-gettext --enable-threads=posix --libdir=/usr/lib --enable-nls --with-sysroot=/ --enable-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-time=yes --with-default-libstdcxx-abi=new --enable-gnu-unique-object --disable-vtable-verify --enable-libmpx --enable-plugin --enable-default-pie --with-system-zlib --with-target-system-zlib --enable-objc-gc=auto --enable-multiarch --disable-werror --with-arch-32=i686 --with-abi=m64 --with-multilib-list=m32,m64,mx32 --enable-multilib --with-tune=generic --enable-offload-targets=nvptx-none --without-cuda-driver --enable-checking=release --build=x86\_64-linux-gnu --host=x86\_64-linux-gnu --target=x86\_64-linux-gnu  Thread model: posix  gcc version 7.3.0 (Ubuntu 7.3.0-27ubuntu1~18.04)  dmitry@dmitry-VirtualBox:~/lubs/12$ ./12.out  01  10  dmitry@dmitry-VirtualBox:~/lubs/12$ ./12.out  0114  1140  dmitry@dmitry-VirtualBox:~/lubs/12$ ./12.out  0002  2000  dmitry@dmitry-VirtualBox:~/lubs/12$ ./12.out  0003 0004  3000 4000  dmitry@dmitry-VirtualBox:~/lubs/12$ ./12.out  003 567 909  300 567 909  dmitry@dmitry-VirtualBox:~/lubs/12$ ./12.out  003003 123 055604  300300 123 556040 |

1. **Дневник отладки** должен содержать дату и время сеансов отладки, и основные ошибки (ошибки в сценарии и программе, не стандартные операции) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Лаб.  или  дом. | Дата | Время | Событие | Действие по исправлению | Примечание |
| 1 | Дом. | 18.11.18 | 20.10 | Не выводит последнее число | Программа выходила из цикла, не выводя последнее число, добавил проверку и вывод последнего числа вне цикла |  |

1. Замечание автора по существу работы
2. Выводы Выполняя данную лабораторную работу студенты могут улучшить свои навыки в программировании и отладки программ, выполняющих обработку чисел в десятеричной системе счисления, могут понять, как устроены, а также как работать с различными целочисленными типами данных не только на языке Си, но и на других языках.

Недочеты, допущенные при выполнении задания, могут быть устранены следующим образом

Подпись студента