SouthWest University

Lab report

Couse name C programming

Semester 2019 - 2020 - 2

Grade 2019 class Software Engineering 3

student Reg. No 222019321062074

Tutor James Chen

School of Computer and Information Science

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Lab 1 | | | **Practicing on conditional statements and loops.** | | | | | |
| Issue Date | | 2020年3月15日 | | | experimental types | | □validation experiment,  □comprehensive experiment  ☑design experiment | |
| Goal  • You will practice using conditional statements and while loops.  Assignment   1. Write a program that calculates and prints the result of the expression as follows 1-1/2+1/3-1/4+…+1/99-1/100. 2. Write a program that is capable of finding the largest and smallest integer in a list of 10 integers.   As input, the program is to read the data form the keyboard.  Write the output to the screen in the form:  **The largest value in the 10 numbers is:**  **The smallest value in the 10 numbers is:**  3. Write a program that uses a while loop to produce this pattern:    4 . Write a program that uses a For loop to produce Chinese multipliction table | | | | | | | | |
| * experimental contents and process  1. Analyze what the problems say, and understand what it means. 2. Clear object procedure. 3. Compilation object process. 4. Make the program functional . | | | | | | | | |
| * screen shots of the VC++ or other IDE for C(Dev c++，VS2010. etc)showing the output results of running your Lab code. | | | | | | | | |
| * Lab Code   Ex1：  #include <stdio.h>  int main (void)  {  double res = 0;  int i;  for (i = 1; i <= 100; i++)  {  if (i % 2)  res += 1.0 / i;  else  res -= 1.0 / i;  }  printf("1-1/2+1/3-1/4+…+1/99-1/100 = %lf", res);  return 0;  }  Ex2：  #include <stdio.h>  #include <limits.h>  int main (void)  {  int i, max = INT\_MIN, min = INT\_MAX, num;  for (i = 1; i <= 10; i++)  {  scanf("%d", &num);  if (num < min)  min = num;  if (num > max)  max = num;  }  printf("The largest value in the 10 numbers is:%d\n", max);  printf("The smallest value in the 10 numbers is:%d\n", min);  return 0;  }  Ex3：  #include <stdio.h>  int main (void)  {  int i, j;  for (i = 1; i <= 10; i++)  {  for (j = 9; j >= i; j--)  printf(" ");  for (j = 2; j <= i; j++)  printf("\* ");  for (j = 1; j <= i; j++)  printf("\* ");  printf("\n");  }  return 0;  }  Ex4：  #include <stdio.h>  int main (void)  {  int i, j;  for (i = 1; i <= 9; i++)  {  for (j = 1; j <= 9; j++)  {  if (i >= j)  printf("%d\*%d=%-2d ", i, j, i \* j);  }  printf("\n");  }  return 0;  } | | | | | | | | |
| * experimental summary/Analysis   Through this experiment, I got a good understanding of loop-structure, and mastered to use the ‘for’ statement, ‘while’ statement and ‘do-while’ statement. Moreover I also knew how to use two or more loop-structure to write a nested loop.  In ex1, ex2 and ex 4, I combined brunch-structure and loop-structure to reach the goals, like using the combination to judge what kind of operation should the program do.  Ex3 doesn’t use brunch-structure, but need careful observation and debugging, which take me a lot time.  In ex2, I used two constant ‘INT\_MAX’ and ‘INT\_MIN’ in <limits.h> to initialize the two variables to avoid error. | | | | | | | | |
|  | Criteria | | | | | | | scale |
| Goal | | | | | | | A B C D E |
| Process | | | | | | |
| Design | | | | | | |
| Algorithm | | | | | | |
| Code | | | | | | |
| Data/Results | | | | | | |
| summary | | | | | | |
| written | | | | | | |
| Score | | |  | | tutor Signature：  Date: : | | |
| * Lab Evaluation Criteria   A: This lab is exceptional, working and meeting all of the specifications.The code is exceptionally well organized and very easy to follow.The code could be reused as a whole or each routine could be reused.The documentation is well written and clearly explains what the code is accomplishing and how.The program was delivered on time.The code is extremely efficient without sacrificing readability and understanding.  B: This lab is very good--works and produces the correct results and displays them correctly. It also meets most of the other specifications. The code is fairly easy to read. Most of the code could be reused in other programs. The documentation consists of embedded comment and some simple header documentation that is somewhat useful in understanding the code. The program was delivered within a week of the due date. The code is fairly efficient without sacrificing readability and understanding.  C: This lab is adequate, with only minor deficiencies. The program produces correct results but does not display them correctly. The code is readable only by someone who knows what it is supposed to be doing. Some parts of the code could be reused in other programs. The documentation is simply comments embedded in the code with some simple header comments separating routines. The code was within 2 weeks of the due date. The code is brute force and unnecessarily long..  D: This lab shows some effort but has at least one major deficiency.The program is producing incorrect results. The code is poorly organized and very difficult to read. The code is not organized for reusability. The documentation is simply comments embedded in the code and does not help the reader understand the code. The code was more than 2 weeks overdue. The code is huge and appears to be patched together.  E: This lab is poorly written and shows very little effort or understanding. | | | | | | | | |