**ENGG1340 Computer Programming II**

**COMP2113 Programming Technologies**

**Module 10 Checkpoint Exercises**

**Name: Zhao Kexin**

**University ID: 3035845368**

**Checkpoint 10.4**

**To answer all the questions, you need to read the external links in the lecture notes first.**

1. Suggest two ways to debug a program.

**Using a Debugger (e.g., GDB)Adding Print Statements**

1. A student wants to use the GDB debugger to debug a c++ program but the debugger fails to tell which line of the code causes the error. Identify the problem and suggest a solution to fix it.

|  |
| --- |
| $ g++ -std=c++11 main.cpp -o main |
| $ ./main  Segmentation fault (core dumped) |
| $ gdb main |
| (gdb) r  Starting program: /home/research/ra/1801/cklai/main  Program received signal SIGSEGV, Segmentation fault.  0x00005555555547d2 in main () |

Not compiling the code with debug information.

$ g++ -g -std=c++11 main.cpp -o main

1. Give one advantage and one disadvantage of inserting cout statements to debug a program.

Advantage:

Quick and Simple: Using cout statements is an easy and fast way to trace program execution and inspect variable values at different points. It doesn't require setting up a complex debugging environment.

Disadvantage:

Clutter and Potential for Mistakes: Inserting too many print statements can clutter the output and make it difficult to interpret. Additionally, these statements need to be removed later, which can be tedious and error-prone, especially in large programs.

1. What is a breakpoint in the GDB debugger? How do you set a breakpoint to a program?

Breakpoint: A breakpoint is a marker placed at a specific line of code in a program where the debugger will pause execution. When the program reaches that point, execution halts, allowing the programmer to inspect the state of the program, including variable values and the call stack.

Setting a Breakpoint: set a breakpoint in GDB using the b command, followed by the line number or function name where you want the program to pause.

1. GDB debugger is used to debug the following program. Suppose a breakpoint is set to line 12. When the program runs, it will pause at line 12. What are the values of n1 and n2 when the program pauses at the first time? Fill in the blank.

|  |
| --- |
| $ cat gcd.cpp  #include <iostream>  using namespace std;  int main()  {  int n1 = 32;  int n2 = 8;  while(n1 != n2)  {  if(n1 > n2)  n1 -= n2; // This line is line 12  else  n2 -= n1;  }  cout << "GCD = " << n1;  return 0;  } |
| $ g++ -g gcd.cpp -o gcd |
| $ gdb gcd |
| (gdb) b 12  Breakpoint 1 at 0x870: file gcd.cpp, line 12. |
| (gdb) r  Starting program: /home/research/ra/1801/cklai/gcd  Breakpoint 1, main () at gcd.cpp:12  12 n1 -= n2; |
| (gdb) p n1  $1 = \_24\_\_\_\_ |
| (gdb) p n2  $2 = \_\_8\_\_\_ |

1. Suppose a breakpoint is set to a certain line in a for loop in a program. The program pauses at that breakpoint when it runs on the debugger. Instead of entering the ‘n’ command multiple times, what command allows the program to resume execution and to pause at the same breakpoint in the next iteration?

The continue command in GDB resumes the program's execution and pauses again at the same breakpoint during the next iteration or when the breakpoint condition is met again.

1. The following C program is to find the sum of Natural Numbers but it contains error(s). Fix the error(s) by modifying the code.

|  |
| --- |
| #include <stdio.h>  int main() {  int n;  int i;  int sum = 0;  printf("Enter a positive integer: ");  scanf("%d", n);  for (i = 1; i <= n; ++i) {  sum += i;  }  printf("Sum = %d", &sum);  return 0;  }  #include <stdio.h>  int main() {  int n;  int i;  int sum = 0;  printf("Enter a positive integer: ");  scanf("%d", &n); // Fix: pass the address of n  for (i = 1; i <= n; ++i) {  sum += i;  }  printf("Sum = %d", sum); // Fix: print sum directly (not as a reference)  return 0;  } |