UNIVERSITY OF THE WESTERN CAPE
DEPARTMENT OF COMPUTER SCIENCE

COS311 2025

Term 2 _ Practical 1 [20 marks]

Final Mark

This practical involves playing with a number of C programs to solve some problems involving pointers. There are 3 different problems.

The exercises here should help make sure you understand all about pointers. It's important to make sure you understand things (get help if you are stuck!).

Student No.	Name	Surname

Exercise 1 – p1.c

Mark
5

Write a short C program that declares and initializes (to any value you like) a double, an int, and a char. Next declare and initialize a pointer to each of the three variables. Your program should then print the address of, and value stored in, and the memory size (in bytes) of each of the six variables.

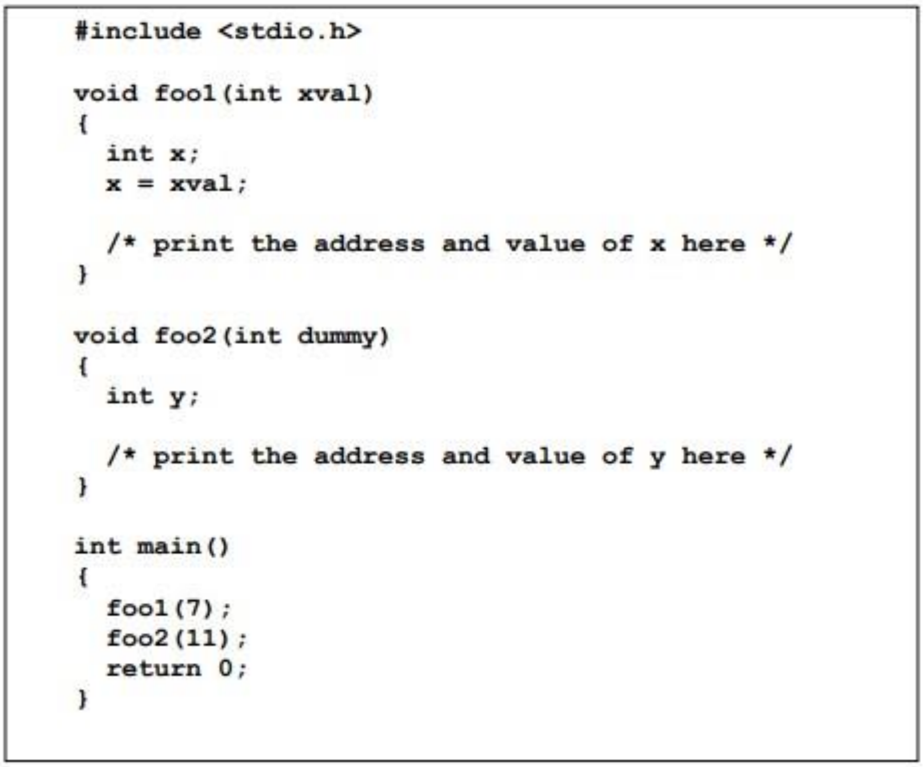
Use the “0x%x” formatting specifier to print addresses in hexadecimal. You should see addresses that look something like this: "0xbfe55918". The initial characters "0x" tell you that hexadecimal notation is being used; the remainder of the digits give the address itself.

Use the sizeof operator to determine the memory size allocated for each variable.

Exercise 2 – p2.c

Mark
5

Find out (add code to print out) the address of the variable x in foo1, and the variable y in foo2. What do you notice? Can you explain this?



```
#include <stdio.h>

void foo1(int xval)
{
  int x;
  x = xval;

  /* print the address and value of x here */
}

void foo2(int dummy)
{
  int y;

  /* print the address and value of y here */
}

int main()
{
  foo1(7);
  foo2(11);
  return 0;
}
```

Exercise 3 – p3.c

Mark
10

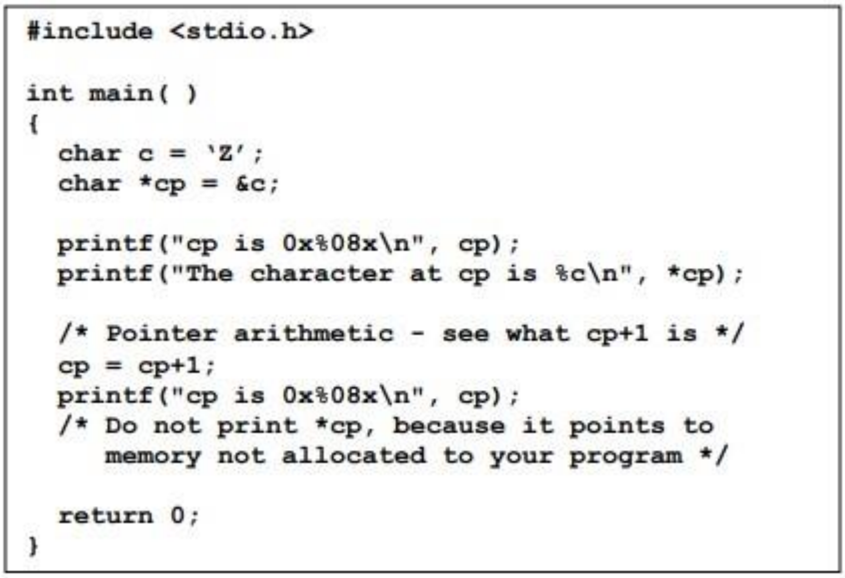
The program below uses pointer arithmetic to determine the size of a 'char' variable. By using pointer arithmetic we can find out the value of 'cp' and the value of 'cp+1'. Since cp is a pointer, this addition involves pointer arithmetic: adding one to a pointer makes the pointer point to the next element of the same type.

For a pointer to a char, adding 1 really just means adding 1 to the address, but this is only because each char is 1 byte.

1. Compile and run the program and see what it does. **(2 Marks)**

2. Write some code that does pointer arithmetic with a pointer to an int and determine how big an int is before and after adding a 1. **(4 Marks)**

3. Same idea – figure out how big a double is, by using pointer arithmetic and printing out the value of the pointer before and after adding a 1. **(4 Marks)**



```
#include <stdio.h>

int main( )
{
  char c = ‘Z’;
  char *cp = &c;

  printf("cp is 0x%08x\n", cp);
  printf("The character at cp is %c\n", *cp);

  /* Pointer arithmetic - see what cp+1 is */
  cp = cp+1;
  printf("cp is 0x%08x\n", cp);
  /* Do not print *cp, because it points to
     memory not allocated to your program */

  return 0;
}
```