Experiment 1

Introduction to pointers and structures in C

1.1 Objectives

The objectives of this lab are:

- ✓ Understanding of syntax of C syntax.
- ✓ Understanding pointers and structures.
- ✓ Solving problems using pointers and structures.

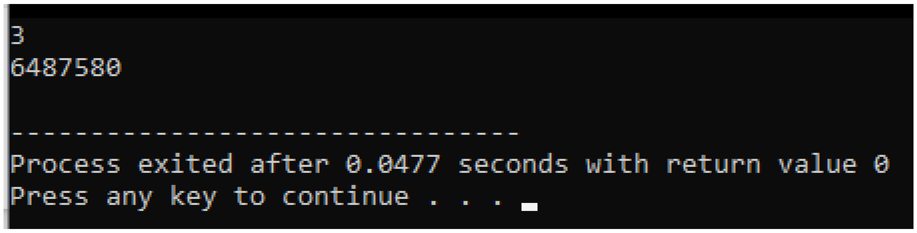
1.2 Introduction

Memory Address:

When a variable is created in C, a memory address is assigned to the variable. The memory address is the location of where the variable is stored on the computer. When we assign a value to the variable, it is stored in this memory address. To access it, use the reference operator (&), and the result represents where the variable is stored:

Example:

```
#include <stdio.h>
int main (void) {
int x = 3;
printf("%d\n", x);
printf("%d\n", &x);
return 0;
}
```



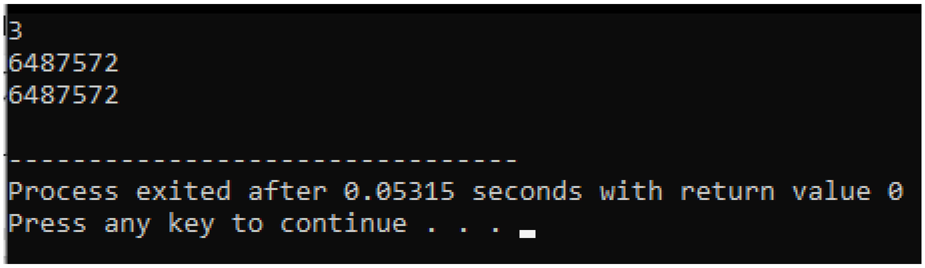
```
3
6487580

--------------------------------
Process exited after 0.0477 seconds with return value 0
Press any key to continue . . .
```

Creating Pointers

A pointer is a variable that stores the memory address of another variable as its value. A pointer variable points to a data type (like int) of the same type, and is created with the * operator. The address of the variable you are working with is assigned to the pointer:

Example:

```
#include <stdio.h>
int main (void) {
int x = 3;
int* ptr = &x;
printf("%d\n", x);
printf("%d\n", &x);
printf("%d\n", ptr);
return 0;
}
```



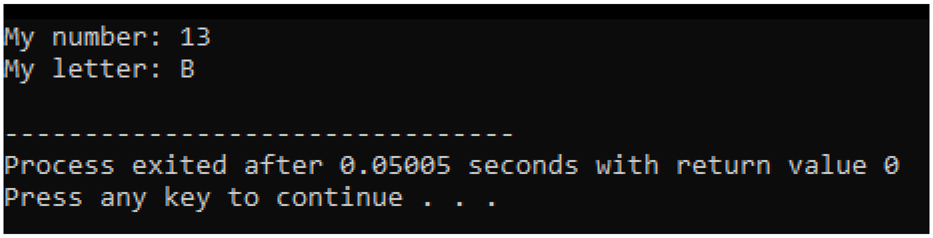
```
3
6487572
6487572

--------------------------------
Process exited after 0.05315 seconds with return value 0
Press any key to continue . . .
```

Structures

Structures (also called structs) are a way to group several related variables into one place. Each variable in the structure is known as a member of the structure. Unlike an array, a structure can contain many different data types (int, float, char, etc.).

Example:

```
#include <stdio.h>
// Create a structure called myStructure
struct myStructure {
  int myNum;
  char myLetter;
};
int main() {
  // Create a structure variable of myStructure called s1
  struct myStructure s1;
  // Assign values to members of s1
  s1.myNum = 13;
  s1.myLetter = 'B';
  // Print values
  printf("My number: %d\n", s1.myNum);
  printf("My letter: %c\n", s1.myLetter);
  return 0;
}
```



```
My number: 13
My letter: B

--------------------------------
Process exited after 0.05005 seconds with return value 0
Press any key to continue . . .
```

1.3 Pre-Lab

1. Lecture notes along with the reference material advised by the instructor should be available during the lab sessions
2. Knowledge to create projects in Microsoft Visual Studio and basic C syntax

1.4 Equipment

- Computer / laptop
- Microsoft Visual Studio C

1.5 Lab Task

Develop a simple payroll application for a company; there are three kinds of employees in the system: salaried employee, hourly employee, and commissioned employee. The system should take input as an array containing employee data and calculates salary.

- The structure Employee has two data variables (name and taxRate).
- The structure Salaried Employee has one data variable (salary).
- The structure Hourly Employee has two data variables (hours and hourlyRate).
- The structure Commissioned Employee has two data variables (sales and commissionRate).

Write the C code for the given problem.

Show the Output of the program

1.6 Procedure

- Create a structure Employee that has two data variables (name and taxRate) and a pointer to a function
- Create a structure Salaried Employee that inherits structure Employee and has one data variable (salary).
- Create a structure Hourly Employee that inherits structure Employee and has two data variables (hours and hourlyRate).
- Create a structure Commissioned Employee that inherits structure Employee and has two data variables (sales and commissionRate).

- Create three functions for calculating salaries for each type.
- Initialize the structures in main function, and while initializing give pointers to the functions for the respective type.

1.7 Lab Report

A printed lab report (10 - 15 pages including title page) is required individually and is due before 4pm one week after the lab. Submit report in the respective lab.

1.8 Results

Show the simulation/ compilation results to the instructor.