Department of ICT
Faculty of Technology
University of Ruhuna

Data Structures and Algorithms – ICT2113 **Level2- Semester -1**

Laboratory Assignment 1 **| 2022**

Objectives

Main objective of this lab session is to revise the knowledge of Arrays, Functions, struct and Pointers.

1.

i. Write a following C program and save it as ***ar1.c***. Compile and run the program

```
#include <stdio.h>
 int main()
{   const int MAX_SIZE=10;
    int arr[MAX_SIZE]; //Declares an array of MAX_SIZE
    int i;
     /* Reads size and elements in array      */
    printf("Enter %d elements in the array : ");
    for(i=0; i< MAX_SIZE; i++)
    {
       scanf("%d", &arr[i]);
    }
     /* Prints all elements of array      */
    printf("\nElements in array are: ");
    for(i=0; i<MAX_SIZE; i++)
    {
       printf("%d, ", arr[i]);
    }
    return 0;
}
```

ii. Modify above program to get summation of numbers and print it.

iii. Create an array of **char type** which can use to store your name.

   a. Store your name on that array.
   b. Print your name using the elements of the array.

2. The following example uses three integers, which are stored in an array of pointers Type the program using g-editor and compile and run. Observe the output

```
#include <stdio.h>

const int MAX = 3;

int main () {

  int  var[] = {10, 100, 200};
  int i, *ptr[MAX];

  for ( i = 0; i < MAX; i++) {
    ptr[i] = &var[i]; /* assign the address of integer. */
  }

  for ( i = 0; i < MAX; i++) {
    printf("Value of var[%d] = %d\n", i, *ptr[i] );
  }

  return 0;
}
```

3. Write a C program according to the following guidelines.
   - Declare **integer, float** and **character** variable called **num, fl** and **ch** respectively.
   - Declare **integer, float** and **character pointer variables** called **numptr, flptr** and **chptr.**
   - Assign values **154, 78.5** and '**g**'to variables **num, fl** and **ch** respectively.
   - Assign address of **num** to **numptr**.
   - Assign address of **fl** to **flptr.**
   - Assign address of **ch** to **chptr**.
   - Print the value of ***numptr, numptr** and **&numptr**
   - Print the value of ***flptr, flptr** and **&flptr**.
   - Print the value of ***chptr, chptr** and **&chptr**.
   - Now add **(*chptr)++;** to your program and print the value of **ch**
   - Do this to other two pointer variables and print the value of **num** and **fl.**

4. Type the following code segment and save it as student.c. Compile it and run. Understand the code with the output.

```
#include <stdio.h>
#define size 3

typedef struct _student {
            char name[50];
             int mark;
   } student;

void print_list(student list[]);
void read_list(student list[]);

main(){
        student list[size];
        read_list(list);
        print_list(list);
        return 0;
}

void read_list(student list[]){
        int i;
        printf("Please enter the student information:\n");
        for(i = 0; i < size;i++)   {
                printf("Name and the marks:");
                scanf("%s %d",list[i].name,&list[i].mark);
        }
}

void print_list(student list[]){
        int i;
        printf("Students' information:\n");
        for(i = 0; i < size;i++)  {
                printf("name: %s, mark: %d\n",list[i].name,list[i].mark);
        }
}
```

5. Try this code segment and understand how pointers are used with structures.

```
#include <stdio.h>

typedef struct AA{
        int x;
} AA;

int main(){
        AA structure;
        AA *ptr;
        structure.x = 46
        ptr = &structure; // & is needed when dealing with structures

        printf("x is=%d\n",ptr->x);
        return 0;
}
```

6. Consider the following code sample and analyze the answers.

```
#include <stdio.h>
void twice (int * val);

int main()
{
 /* & = "Address of …..''  *="Content of …..  "      */
  int x;      int *y;
   x=56;     y=&x;
   twice(&x);
   printf("x value=%d\n",x);
   printf("y  memory address= %p\n",y);
   printf('and  value of y = %d\n",*y);
}
   void twice (int *val)
{
 *val=*val*2;  }
```

7. Write two functions to interchange two integers. One function should pass the parameters by reference (**pchange())**while the other one should pass the parameters by values (**change())**. Call the function from main function and check the answers.

```
void change(int x, int y);
void pchange(int *a, int *b);
```

8. Write a C program to perform the following.
   a) Define a structure called book that describes the following information of some C programming books: title, author, ISBN number and price.
   b) Declare an array to store the details of 10 books.
   c) Input the details of the books from the keyboard.
   d) Write a separate function to display the titles of the books, prices of which are below Rs. 2000.00.

   Suppose a customer is interested in buying books written by the author "Kernighan". Write a separate function to display the titles and the prices of the books written by the above author.

9. Following program is written to referencing pointer to another address to access the memory. Write the program, compile and run. Observe the output.

```
#include <stdio.h>
typedef struct person
{
  int age;
  float weight;
};

int main()
{
   struct person *personPtr, person1;
   personPtr = &person1;          // Referencing pointer to memory address of person1

   printf("Enter integer: ");
   scanf("%d",&(*personPtr).age);

   printf("Enter number: ");
   scanf("%f",&(*personPtr).weight);

   printf("Displaying: ");
   printf("%d%f",(*personPtr).age,(*personPtr).weight);

   return 0;
}
```

   Above program, the pointer variable of type struct person is referenced to the address of person1. Then, only the structure member through pointer can can accessed.

   Using -> operator to access structure pointer member

   Structure pointer member can also be accessed using -> operator.

(*personPtr).age is same as personPtr->age
(*personPtr).weight is same as personPtr->weight

Modify the same program by adding -> operators.

10. Following C program demonstrates example of structure pointer using user define function.

```
struct item
{
      char  itemName[30];
      int qty;
      float price;
      float amount;
};

/*readItem()- to read values of item and calculate total amount*/
void readItem(struct item *i)
{
      /*read values using pointer*/
      printf("Enter product name: ");
      gets(i->itemName);
      printf("Enter price:");
      scanf("%f",&i->price);
      printf("Enter quantity: ");
      scanf("%d",&i->qty);

      /*calculate total amount of all quantity*/
      i->amount =(float)i->qty * i->price;
}

/*printItem() - to print values of item*/
void printItem(struct item *i)
{
      /*print item details*/
      printf("\nName: %s",i->itemName);
      printf("\nPrice: %f",i->price);
      printf("\nQuantity: %d",i->qty);
      printf("\nTotal Amount: %f",i->amount);

}
int main()
{
```

```
    struct item itm;          /*declare variable of structure item*/
    struct item *pItem;       /*declare pointer of structure item*/

    pItem = &itm;     /*pointer assignment - assigning address of itm to pItem*/
    /*read item*/
    readItem(pItem);
    /*print item*/
    printItem(pItem);


    return 0;
}
```