भारतीय सूचना प्रौद्योगिकी संस्थान गुवाहाटी

Indian Institute of Information Technology Guwahati

COMPUTER PROGRAMMING LAB (CS110)
ASSIGNMENTS–11

1. Write a macro to test whether a character is a small case letter or not.
2. Write a macro to find the larger number of two numbers.
3. Print the source filename, date of compilation, time of compilation, function name, and line number using a macro.
4. Create a structure `student` in C to store the following information about a student:
   i. `name`, a string (an array) of 11 characters.
   ii. `roll`, an integer.
   iii. `sex`, a character, 'M' (male), 'F' (female), 'T' (third gender/ other).
   iv. `gpa`, i.e., grade point average, a real (`double`) value.

   Now, perform the following:
   i. Write a function to print an instance of the structure. You need to pass an instance of `student` to the function. You need to use the "`.`" operaotor to access the member variables.
   ii. Create an instance/ object of `student`. Print the address of the instance. Now, print the address of each of its member variables. Print the size of the structure using the `sizeof()` operator.
   iii. Use the preprocessor directive
   `#pragma pack(1)`
   before defining the structure. Create an instance/ object of `student`. Print the address of the instance. Now, print the address of each of its member variables. Print the size of the structure using the `sizeof()` operator.
   iv. Create an array of five `student` objects taking user inputs. Print the details of each student. Print the address of each of the five objects.

v. Write a function to print an instance of the structure. You need to pass the pointer of an instance of `student` to the function. You need to use the "`->`" operaotor to access the member variables.

5. Realize the following program:

```c
#include <stdio.h>
typedef struct node_t {
    int data;
    struct node_t *next;
} Node_t, *Node;
void f(Node_t *h) {
    h ? printf("%d -> ", h->data), f(h->next) : printf("NULL");
}
void g(Node_t *h) {
    h ? g(h->next), printf(" <- %d", h->data) : printf("NULL");
}
int main() {
    Node_t n4 = {4, 0}, n3 = {3, &n4}, n2 = {2, &n3}, n1 = {1, &n2};
    Node h = &n1;
    f(h);
    printf("\n");
    g(h);
    return 0;
}
```

6. Define a `union` that contains (i) a `char` variable, (ii) an `int` variable, (iii) a `float` variable, and (iv) a `double` variable. Now, perform following:

   i. Create an object of the `union`. Print the address of each variable.

   ii. Print the size of the `union` using the `sizeof()` operator.

   iii. Create an object of the `union` and initialize it to zero (use "`= {0}`" during initialization). Assign a value to the `char` variable and print the other member variables.

   iv. Create an object of the `union` and initialize it to zero (use "`= {0}`" during initialization). Assign a value to the `int` variable and print the other member variables.

   v. Use a pointer to the created object. Now, access the elements using the "`->`" operator.

7. Define a structure `S` that has two member variables: (i) a member of type `int` and (ii) a member that of a nested-structure, `P`. `P` has two member variables: (i) a member variable of type `char`, and (ii) a member variable of a nested-union `U`. `U` has a member of type `char` and a member of type float. Create an object of this structure. Scan each of these member variables from the keyboard. Print each of these member variables.