



भारतीय सूचना प्रौद्योगिकी संस्थान गुवाहाटी
Indian Institute of Information Technology Guwahati
COMPUTER PROGRAMMING LAB (CS110)
SOLUTIONS-11

1. Write a macro to test whether a character is a small case letter.

```
#include <stdio.h>

#define IS_SMALL_LETTER(c) \
    ((c) >= 'a' && (c) <= 'z')

int main() {
    char c = 'C';
    printf("%d\n", IS_SMALL_LETTER(c));
    return 0;
}
```

2. Write a macro to find the larger number of two numbers.

```
#include <stdio.h>

#define MAX(a, b) \
    ((a) > (b) ? (a) : (b))

int main() {
    int a = 100, b = 20;
    printf("%d\n", MAX(a, b));
    return 0;
}
```

3. Print the source filename, date of compilation, time of compilation, function name, and line number using a macro.

```
#include <stdio.h>

int main() {
    printf("Source File Name: %s\n", __FILE__);
    printf("Date of Compilation: %s\n", __DATE__);
    printf("Time of Compilation: %s\n", __TIME__);
    printf("Function Name: %s\n", __func__);
    printf("Line Number: %d\n", __LINE__);
}
```

```
    return 0;
}
```

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 must pass an instance of student to the function. You must access the member variables using the "." operator.
- 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 structure size 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 structure size 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 must pass the pointer of an instance of student to the function. You must use the "->" operator to access the member variables.

```
#include <stdio.h>

//#pragma pack (1) //Uncomment this

struct Student {
    char name[11];
    int roll;
    char sex;
    double gpa;
};

void printStudent(struct Student s) {
    printf("Name: ");
    puts(s.name);
}
```

```

    printf("Roll: %d\n", s.roll);
    printf("Sex: %c\n", s.sex);
    printf("GPA: %lg\n", s.gpa);
}

void printStudent2(struct Student *ps) {
    printf("Name: ");
    puts(ps->name);
    printf("Roll: %d\n", ps->roll);
    printf("Sex: %c\n", ps->sex);
    printf("GPA: %lg\n", ps->gpa);
}

void scanStudent(struct Student *ps) {
    printf("Enter Name: ");
    gets(ps->name);
    fflush(stdin);
    printf("Enter Roll: ");
    scanf("%d", &ps->roll);
    fflush(stdin);
    printf("Enter Sex: ");
    scanf("%c", &ps->sex);
    printf("Enter GPA: ");
    scanf("%lg", &ps->gpa);
    fflush(stdin);
}

int main() {
    struct Student s = {
        "Name", 27, 'M', 9.0
    };
    printStudent(s);
    printf("&s      = %p (%d)\n", &s, &s);
    printf("&s.name  = %p (%d)\n", &s.name, &s.name);
    printf("&s.roll   = %p (%d)\n", &s.roll, &s.roll);
    printf("&s.sex    = %p (%d)\n", &s.sex, &s.sex);
    printf("&s.gpa    = %p (%d)\n", &s.gpa, &s.gpa);
    printf("sizeof(s) = %d\n", sizeof (s));
    int n = 5;
    struct Student students[n];
    for (int i = 0; i < n; i++) {
        scanStudent(students + i);
    }
    for (int i = 0; i < n; i++) {
        printStudent(students[i]);
    }
    return 0;
}

```

5. Realize the following program:

```

#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 the 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. Access the elements using the "->" operator.

```

#include <stdio.h>

union U {
    char c;
    int i;
    float f;
    double d;
};

int main() {
    union U u = {0};
    printf("&u      = %p (%d)\n", &u, &u);
    printf("&u.c    = %p (%d)\n", &u.c, &u.c);
}

```

```

printf("&u.i      = %p (%d)\n", &u.i, &u.i);
printf("&u.f      = %p (%d)\n", &u.f, &u.f);
printf("&u.d      = %p (%d)\n", &u.d, &u.d);
printf("sizeof(u) = %d\n", sizeof (u));
u.c = 'a';
printf("u.c      = %c" "\n", u.c);
printf("u.i      = %d" "\n", u.i);
printf("u.f      = %g" "\n", u.f);
printf("u.d      = %lg" "\n", u.d);
union U v = {0};
v.i = 100;
printf("v.c      = %c" "\n", v.c);
printf("v.i      = %d" "\n", v.i);
printf("v.f      = %g" "\n", v.f);
printf("v.d      = %lg" "\n", v.d);
union U *p = &v;
p->c = 'x';
p->i = 10;
p->f = 3.5;
p->d = 3.9;
return 0;
}

```

7. Define a structure S that has two member variables: (i) a member of type int and (ii) a member 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.

```

#include <stdio.h>

struct S {
    int i;
    struct P {
        char c;
        union U {
            char c;
            float f;
        } u;
    } p;
    double d;
};

void scanS(struct S *ps) {
    printf("Enter i: ");
    scanf("%d", &ps->i);
    fflush(stdin);
    printf("Enter p.c: ");
    scanf("%c", &ps->p.c);
}

```

```

    printf("Enter 1 to enter u.c and any other value for u.f: ");
    int choice = 0;
    scanf("%d", &choice);
    fflush(stdin);
    if (choice == 1) {
        printf("Enter p.u.c: ");
        scanf("%c", &ps->p.u.c);
    } else {
        printf("Enter p.u.f: ");
        scanf("%g", &ps->p.u.f);
    }
}

void printS(struct S *ps) {
    printf("i: %d\n", ps->i);
    printf("p.c: %c\n", ps->p.c);
    printf("p.u.c: %c\n", ps->p.u.c);
    printf("p.u.f: %g\n", ps->p.u.f);
}

int main() {
    struct S s = {0};
    scanS(&s);
    printS(&s);
    return 0;
}

```