Naseem Auguste

Exercise on structure

A couple of additional notes about structure before starting the exercise:

- The Structure members are stored together in the memory one after another according to the sequence of their declaration inside the structure.
- The best way to know the size of your structure is using the sizeof() function. Just adding the variable sizes inside the structure is not enough to know the size as the operating system can add additional memory for the structure to efficiently access it.
- We can assign a struct variable to another variable of the same type. When we assign a struct variable to another, all members of the variable are copied to the other struct variable. Even if you have an array inside your structure, the assigned structure will have its own copy of the array.

1.) What would be the output of the following programs:

(a)
```
int main( ) {
   struct message{
     int num ;
     char msg1[50] ;
     char msg2[50] ;
   } m ;
  m.num = 1 ;
  strcpy ( m.msg1, "We had lot of homework." );
  strcpy ( m.msg2, " Hope it is the last one);
  /* assume that the strucure is located at
address 2004 */
  printf ( "\n%u %u %u\n", &m.num, m.msg1,
m.msg2 ) ;
  printf ( "\n%d %s %s", m.num,m.msg1,m.msg2) ;
  return 0;
}
```
2004 2008 2012

1 we had lot of homework

Hope it is the last one

b)
```
struct messages
{
        int num ;
        char mess1[50] ;
        char mess2[50] ;
        } m1 = { 2, "If you are driven by
success", "make sure that it is a quality drive"
             } ;
main( )
{
        struct gospel m2, m3 ;
        m2 = m1 ;
        m3 = m2 ;
        printf ( "\n%d %s %s", m1.num,
m2.mess1, m3.mess2 ) ;
}
```
? nothing

```
c)typedef struct X { int num ;  int *y; }X;
int main( ) {
  X x;
  X *p;
  p=&x;
  x.num = 10;
  x.y = &x.num; //consider the address is 104
  printf("%d %d %u %u %u %d %d", p->num,
(*p).num, &x.num, x.y, p->y, *(x.y), *(p->y));
  return 0;

}
```
10 10 104 104 104 10 10

```
d) struct Point {  int x ; int y; };
struct Point add(struct Point p,
int i, int j) {
  p.x += i;
  p.y += j;
  return p;
}
int main( ) {
  struct Point p1 = {0,0};
  p1 = add(p1, 1, 1);
  printf("%d %d", p1.x, p1.y);

}
```
1,1

e)
```
struct Point {  int x ; int y; };

void add1(struct Point p, int i, int j) {
  p.x += i;
  p.y += j;
}

void add2(struct Point *p, int i, int j) {
  p->x += i;
  p->y += j;
}

int main( ) {
  struct Point p1 = {0,0};
  add1(p1, 1, 1);
  printf("%d %d\n", p1.x, p1.y);

  add2(&p1, 1, 1);
  printf("%d %d", p1.x, p1.y);
  return 0;

}
```

0 0

1 1

f)
```
struct Point {  int x ; int y; };
void add(struct Point p[], int a,
int b) {
  for(int i=0; i<2; i++){
      p[i].x += a;
      p[i].y += b;
  }
}
void add2(struct Point *p, int i,
int j) {
  p->x += i;
  p->y += j;
}
int main( ) {
  struct Point p1[2]={0, 0, 0, 0};
  add(p1, 1, 1);

  for(int i=0; i<2; i++) {
      printf("%d %d\n", p1[i].x,
p1[i].y);
  }

  return 0;
}
```

1 1

1 1

2.) Point out the errors, if any, in the following programs:

(a)
```
int main( )
{
        struct employee
        {
                char name[25] ;
                int age ;
                float bs ;
        } ;
        struct employee e ;
        strcpy ( e.name, "Hacker" ) ;
        age = 25 ;
        printf ( "\n%s %d", e.name, age ) ;
       return 0;
}
```

need prefix or data type

(b)
```
int  main( )
{
        struct
        {
                char name[25] ;
                char language[10] ;
        } ;
        struct employee e = { "Hacker",
"C" } ;
        printf ( "\n%s %d", e.name,
e.language ) ;
        return 0;
}
```

no structure variable with it

(c)

```
struct virus
{
        char signature[25] ;
        char status[20] ;
        int size ;
        } v[2] = {
                "Yankee Doodle", "Deadly", 1813,
                "Dark Avenger", "Killer", 1795
                } ;
main( )
{
        int i ;
        for ( i = 0 ; i <=1 ; i++ )
        printf ( "\n%s %s", v.signature, v.status ) ;
}
```

(d)

```
 struct s
{
        int i ;
        struct s *p ;
} ;
int main( )
{
        struct s var1, var2 ;
        var1.i = 100 ;
        var2.i = 200 ;
        var1.p = &var2 ;
        var2.p = &var1 ;
        printf ( "\n%d %d", var1.p -> i, var2.p ->
i ) ;
       return 0;
}
```

no error

(e)

```
struct s
{
        char ch ;
        int i ;
        float a ;
} ;
int main( )
{
        struct s var = { 'C', 100, 12.55 } ;
        f ( var ) ;
        g ( &var ) ;
       return 0;

}
void f ( struct s v )
{
        printf ( "\n%c %d %f", v -> ch, v -> i, v -> a ) ;
}

void g ( struct s *v )
{
printf ( "\n%c %d %f", v.ch, v.i, v.a ) ;
}
```

invalid
operation

3. Short Answer question:

a) When you should use dot (.) operator and when you should use (->) operator to access the structure members?

b) Discuss typedef key word and what happen when you do not use typedef?

c). Show an example of nested structure with example of accessing the members of nested structures.

d). A structure variable is passed by value or reference to a function? Experiment with the relevant output tracing code shown above as an example!

e) A structure array is passed by value or reference to a function? Experiment with the relevant output tracing code shown above as an example!

And don’t forget to go through the examples in the structure slide and at least write few codes on structure (if you have not written much code using structure before).. In the slide, there were some extra problems on the book example, finding the book with maximum price, etc., could be a good practice! Of-course you should start writing those codes from scratch. You don’t have to submit the codes. But it would be a good practice.