Chapter 6 Summary

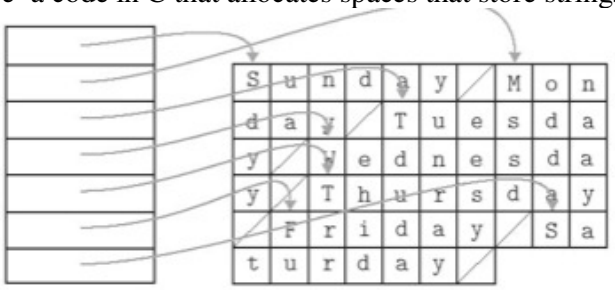
-Understand the representations of primitive types such as int, float, char, string.

-Understand the representations of complex data type, user-defined data type : array, record, union, enumerated type, ordinal type etc.

-Understand pointer problems and solutions at compile time and run time, garbage collection algorithms, pros/cons of each algorithm.

Exercises: Chapter 6

1. What are examples of primitive types?
2. Give an example of ordinal types in C language.
3. There are several ways to represents a string in a language. List two examples of the language you know. What are the pros and cons of the representations?
4. Describe a formula of the address of the rowwise and columnwise array representation.
5. An array in C starts at index 0, while an array in Pascal can start at any index. Describe how the formula of the array address calculation will be different.
6. Declare an enumerate type in C storing a set of value "low, medium, high". Suppose this is the fan speed. For a temperature value, the possible value is "hot, mild, cold". Declare an enumerate type for this. And write the C code to read input temperature and output the proper speed by printing out the proper fan speed. (Hint: you com up with rules say "if the temperature is hot, the fan speed is high")
7. Consider the above question. If we do not use the enumeration, what are other ways to implement it? What is the advantage of using the enumeration? (Hint: consider the pros of using enumeration/subrange.)
8. Give a code in C that allocates spaces that store strings as following:



Let's call the array of points "week"
What is "week[1][2]"? Can we access it right? If not, how to manage the pointers so that we can access "week[1][2]".

9. Compare and contrast between using row pointer and contiguous memory to implement an array.

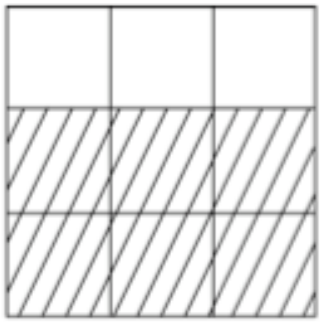
10. What is the main difference between record (structure) and array?

11. What are the possible storage binding of an array and array index?

12. Assume that multidimensional array is stored in a row-major order (1st dimension). Develop an access function of an element for a 3-dimensional array. Assume that M,N, and O are the size of each dimension (e.g., number of rows, columns, heights for the 1st, 2nd, and 3rd dimensions) (i.e. the memory location of `m[I,J,K]` is computed by?)

13. Give a example of C function that declares local arrays where source of array is from stack, the binding of the subscript is dynamic.

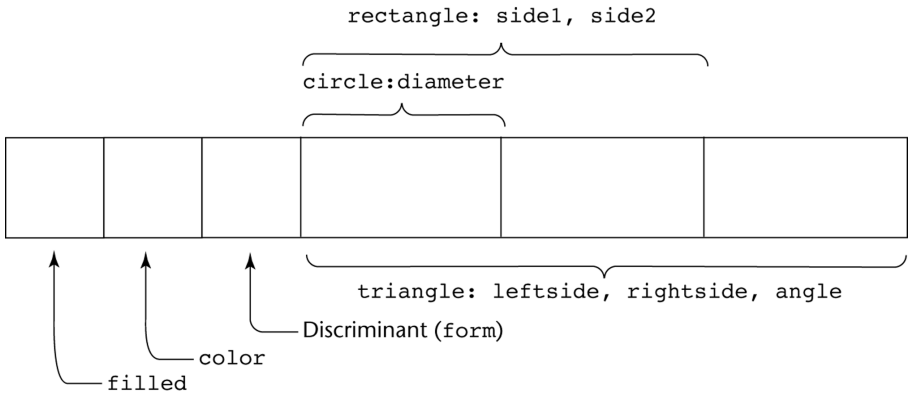
14. Consider the example of slice in Fortran.



MAT (2:3, 1:3)

How to write code in C to refer as this.

15. Consider the following structure.



Assume that side1, side2, leftside, rightside, angle are float, and diameter is double.
Assume filled and color is integer (16 bits).
Assume discriminant is 1 char (1 byte).
(i) declare it is C language. What is the total size of the whole structure?
(ii) Assume float is 32 bits and 1 word is 32 bits.
Draw the structure of the whole record. How many bytes we have for the memory hole when considering the memory is aligned.
(iii) Assume the starting address of this variable, called, myshape, is at 1000. What is the address if "myshape.circle.diameter"? Assume 1 word transfer per cycle. How many cycles needed to get the value?

16. Explain the problems of
    (a) Dangling pointer.
    (b) Lost heap memory
17. Draw the picture of how the values of each variable are changed in a memory based on the following code segment.

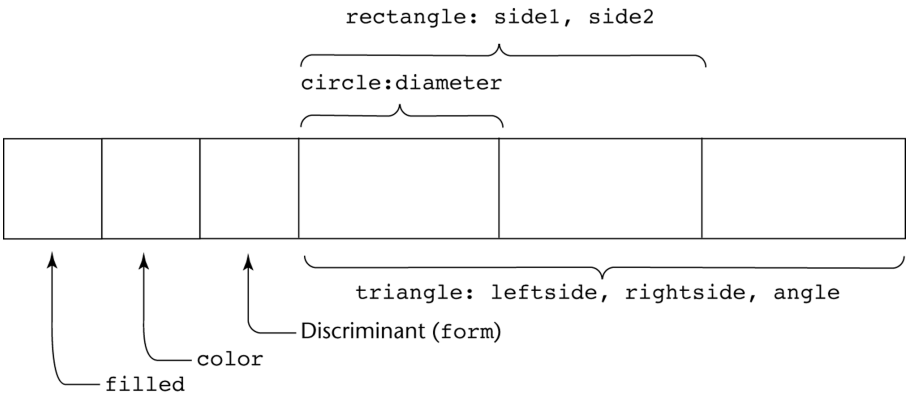
```
int *p1,*p2,*p3;
p1 = (int* )malloc (sizeof (int) *30);
p2 = (int* )malloc (sizeof (int) *40);
p3 = (int* )malloc (sizeof (int) *20);
p3=p1;
p2 = p1;
free(p3);
```

(a) If integer and pointer have size 2 bytes each, how many bytes are lost heap memory? What are dangling pointers?

(b) If we use tomb stone approach to solve dangling pointer, draw a memory configuration of the above code until before the free statement. How many bytes are used for tombstone nodes?

(c) If we use lock and key approach, how many bytes are used to store key items?

18. Write the example of C program that generates a circular linked list which later becomes lost heap of the size at least 1K bytes.
19. Consider the following structure.



Assume that side1, side2, leftside, rightside, angle are float, and diameter is double.
Assume filled and color is integer (16 bits).
Assume discriminant is 1 char (1 byte).

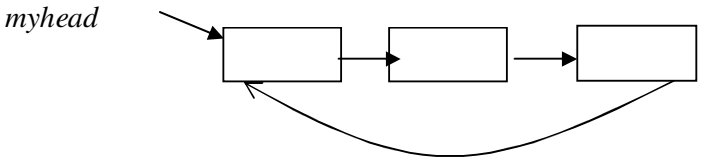
(a) Declare it is C language. What is the total size of the whole structure?

(b) Assume float is 32 bits and 1 word is 32 bits.
Draw the structure of the whole record. How many bytes we have for the memory hole when considering the memory is aligned.

(c) Assume the starting address of this variable, called, myshape, is at 1000. What is the address if "myshape.*circle* variable.diameter"? Assume 1 word transfer per cycle. How many cycles needed to get the value? (Assume the memory is aligned by words.)

20. Answer the following questions.

(a) Write the code that create a circular list like this and set *myhead* point to the first node.



(b) If we add the statement,

```
int *ptr = myhead->next.
```

write the RC value for each node.

(c) If we set

```
ptr=myhead = NULL;
```

Assume pointers are two bytes and integer is one byte. Assume the structure contains 1 key integer and 1 next node pointer.

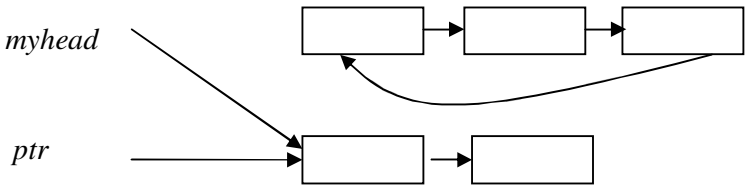
How many bytes are lost in the heap memory?

(d) If we use the tombstone nodes to solve the dangling pointer problem, draw the resulting configuration.

How many bytes are used for tombstone nodes?

Extra: What about lock-and-key approaches? What is the overhead size?

(e) What if we add more nodes, it becomes like the following.



If the heap size is 128 bytes, trace the copy collection algorithm. What are available bytes after the garbage collection? Draw how it looks like after the garbage collection.

21. Consider the following declaration.

```
struct MYREC1
{
      int x;
      float y;
      char c[10];
} m1;

union MYREC2
{
      int x;
      float y;
      char c[10];
} m2;
```

Suppose that int has the size of 2 bytes, float has the size of 4 bytes and char has the size of 1 byte.

(a) Draw the memory storage of variables m1, and m2.

(b) What are the size of m1 as well as m2?

22. Draw a memory storage for the following record. What is the total size of the structure Char is 1 byte , float is 2 bytes int is 2 byte and double is 4 bytes. (5pts)

```
struct  test2
{
       int a;
       char b[10];
       double c;
       union test {
        int x[4];
        float y[2];
        char z;
       } t;
      float k;
} mytest;
```

23. Given the example of the following declaration of array in C,

```
char a[3][3][2] =
{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17};
```

(a) What is the element a[2][1][1] ?
(b) What is the address of a[2][1][1]?

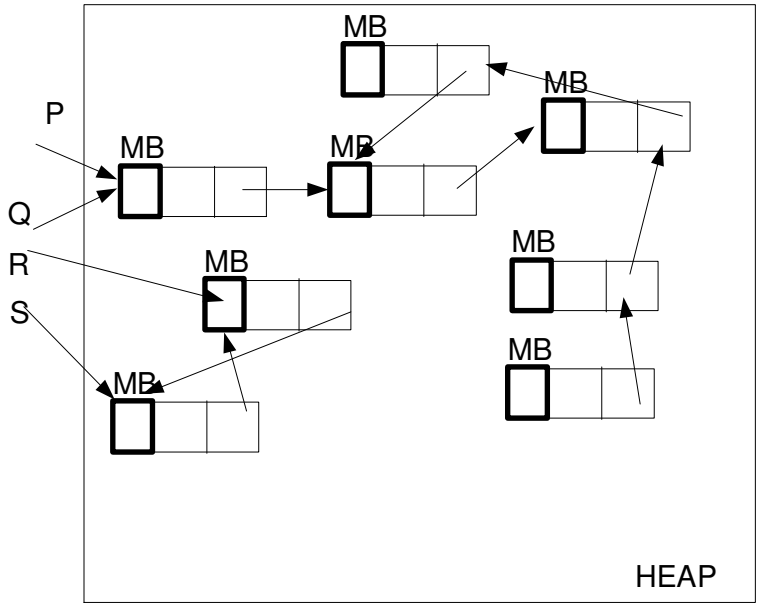
24. Compare and contrast arrays and records.

25. Draw the picture of how each pointer variables are changed in a memory based on the following code segment.

```
struct node
{
        char key[3];
        struct node *next;

};


int main(int argc, char *argv[])
{
         struct node *head, *tail, *p1, *p2;
          head = (struct node *)malloc(sizeof (node) );
          p1=head;
          for (i=0; i < 5; i++) {

          p1->next = (struct node *)malloc(sizeof (node) );
          strcpy(node->key,"123");
          p1 = p1->next;

        } //** 1
           p1->next=NULL
        p2=p1; // ***2
        free(p1);  //** 3
        head = NULL;      //** 4

}
```

If char is of size 1 byte, int is 1 byte and pointer variable is 2 bytes,
(a) What is the size of the node in bytes?
(b) If tombstone node approach is used, at line 1 what is the total size occupied by the tombstone nodes?
Draw a figure when the tombstone node is used after finished line 1.
What about using lock-and-key, how many overhead bytes are needed?
(c) What is the reference count value of each node at line 2.
(d) How many bytes are lost heap memory at line 4?
(e) What are dangling pointers at line 3?

26. Consider the following memory allocation in the heap.



(a) After applying reference counting algorithm, what are the RC values for each memory node? Indicate the value for each node in the field MB.

(b) After applying mark-sweep algorithm, indicate the values of the bit in each memory piece. What is the memory configuration after applying the algorithm? Draw your picture.

(c) After applying copy-collection algorithm, what is the memory configuration? Draw your picture.