• This is an OPEN Book, CLOSED Notes exam.

• Turn off all your electronic devices. NO Devices are allowed.

• You have 50 minutes time.

• This page is the answer page for the multiple choice questions. Mark your answers clearly by checking ( X ) the boxes on this page. Answers marked on other pages will not be considered.

• There are 20 multiple choice questions and 2 free form questions. Make sure you have all the pages with the questions.

• For each question, there are four choices of answers. Mark ( X ) the corresponding box on this answer sheet.

<table>
<thead>
<tr>
<th>Question No.</th>
<th>i</th>
<th>ii</th>
<th>iii</th>
<th>iv</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question No.</th>
<th>i</th>
<th>ii</th>
<th>iii</th>
<th>iv</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. In the following block of code, identify the global variable(s)

```c
#include <stdio.h>
int square(int a);
int x = 5;
int y = 7;

int square(int a)
{ int s;
s = a * a;
return s;
}

int main(void)
{ int z;
z = square(x);
printf("%d\n", z);
return 0;
}
```

- i. x, y
- ii. x
- iii. y
- iv. z

2. In the block of code given in Q1. identify the local variable(s) in main()

- i. a
- ii. s
- iii. x
- iv. z

3. Within a function, which has a local variable with the same name as a global variable,

- i. Global scope takes precedence
- ii. Local scope takes precedence
- iii. Both have equal scope
- iv. Scope depends on when the function is called

4. Which of the symbols is used for logical negation

- i. &&
- ii. !
- iii. ||
- iv. !!

5. Which of the following is incorrect

- i. Enumerator values are integer constants
- ii. By default, enumerator values start at 0 and are incremented by 1 for each following member
- iii. Specific enumerator values may be defined by the user
- iv. Enumerator values are float constants

6. Consider the following block of code

```c
struct my_struct{
int x;
int y;
/
struct my_struct * my_ptr;
```

which of the following is a correct way to allocate memory

- i. my_ptr = (struct my_struct *)malloc(sizeof(struct my_struct));
- ii. my_ptr = (struct my_struct *) malloc(sizeof(struct my_struct));
- iii. my_ptr = (struct my_struct *) malloc(sizeof(struct my_struct));
- iv. my_ptr = (struct my_struct *) malloc(sizeof(struct my_struct));

7. For my_ptr defined in Q6, assuming memory has been allocated correctly
How would you set the structure member x to 6

- i. my_ptr->x = 6
- ii. my_ptr.x = 6
- iii. my_ptr(x) = 6
- iv. my_ptr = (x)6

8. What is the output of the printf in the following block of code

```c
enum Weekday
{
Monday = 2,
Tuesday,
Wednesday,
Thursday,
Friday,
Saturday,
Sunday = 1;
};
```

```c
enum Weekday Today = Wednesday;
```

```c
printf("%d", Today);
```

- i. 1
- ii. 2
- iii. 3
- iv. 4

9. During program execution, local variables are stored in the

- i. stack
- ii. heap
- iii. program segment
- iv. .c source file
10. When you use malloc() to do dynamic memory allocation, memory is allocated out of
   i. stack ii. heap iii. program segment iv. the compiler

11. Assuming that a char takes 1 byte of memory space, for the following block of code

   ```c
   char abc[10];
   printf("%d", sizeof(abc));
   ```

   would produce the output
   i. 10 ii. 1 iii. 3 iv. abc

12. Which of the following is a valid declaration for a pointer struct_ptr, which can point to a variable of type struct my_struct

   i. `struct my_struct * struct_ptr = NULL;`
   ii. `struct my_struct struct_ptr *;`
   iii. `struct my_struct struct_ptr * = NULL;`
   iv. `struct_ptr * struct my_struct;`

13. `int a = 10
    int * b = &a;`

   The value stored in a, can be accessed through b using
   i. *b ii. b* iii. &b iv. b&

14. The binary equivalent of decimal number 19 is

   i. 1000 ii. 10011 iii. 11001 iv. 1001

15. In the block of code given below

   ```c
   int * p ;
   int b ;

   b = 9;

   p = &b;
   ```

   i. p contains the address of b
   ii. p contains the value 9
   iii. p contains the value of b
   iv. p is used incorrectly

16. What is the output of the following block of code

   ```c
   int x[5] = {10,20,10,20,10}; /* array of 5 integers */
   int *p1, *p2;
   p1 = &x[1]; p2 = &x[3];

   if (p1 == p2)
   {
     printf("p1 and p2 are identical!\n");
   }
   else if (*p1 == *p2)
   {
     printf("Contents of p1 and p2 are the same!\n");
   }
   else
   {
     printf("No relationship between p1 and p2\n");
   }
   ```

   i. p1 and p2 are identical!
   ii. Contents of p1 and p2 are the same!
   iii. No relationship between p1 and p2
   iv. None of the above

17. The execution time, for extracting the smallest integer, in a sorted integer array of size n, is

   i. O(n) ii. O(1) iii. O(n*n) iv. O(log(n))

18. Every node of any linked list

   i. Has to contain a string name
   ii. Has to contain an integer key value
   iii. Has to contain a pointer which can point to a similar node
   iv. Has to have a pointer pointing to NULL

19. `struct xzy {
     int a;
     struct xzy * link;
   }`

   Consider a node of type xzy that is currently in a linked list, if the link pointer of the node points to NULL then

   i. The node is definitely the last node in the list
   ii. The node need not be the last node in the list
   iii. The node cannot be the first in the list
   iv. Irrespective of insertion strategy, a new node has to be always inserted after this node
20. In the following block of code

```c
int a;
int * ptr;

a = 9;
c = f(&a);
```

the input parameter for the function f() is passed by

i. value  ii. reference  iii. coercion  iv. shoving

Programming/Program Analysis questions :

21. Write a function that reorders the values in two integer variables, such that the values are in ascending order.
   Assume that the corresponding function prototype statement is
   ```c
   void reorder(int *a, int * b);
   ```
   when your function returns *a should point to the smaller value and *b to the bigger
   Example Usage:
   ```c
   int a = 6;
   int b = 3;
   reorder(&a, &b);

   printf("a = %d b = %d", a, b);
   ```
   The output of this should be
   ```
   a = 3 b = 6
   ```
   Fill in the code for the function , within the definition given below
void reorder(int *a, int * b);
{
    // write your code below this comment
    return;
}

22. Consider the following structure

struct list{
    int key;
    struct list * next;
}

struct list * head;  // global pointer to the first node of a linked list

void func1(int key);
{
    struct list * tmpptr = head;
    if( tmpptr == NULL)
        return;

    // code to process the linked list

    tmpptr = tmpptr->next;
    // process the next node
}

// further processing of the linked list

if (tmpptr->key == key)
{
    head = tmpptr->next;
    free(tmpptr);
    return;
}

while (tmpptr->next != NULL)
{
    if (key == tmpptr->next->key)
    {
        struct list *tmpptr2 = tmpptr->next;
        tmpptr->next = tmpptr->next->next;
        free(tmpptr2);
        return;
    }
    tmpptr= tmpptr->next
}
// end of while

}// end of func1

i. What does the function func1() do?

ii. What would the following linked list look like, after the call func1(24)?

head=>[22,6500]=>[23,2322]=>[24,NULL]

2500 6500 2322

Show the resultant linked list below