Name: ___________________________ Section: ___________________

READ and complete the following:

- Bubble your Scantron only with a No. 2 pencil.
- On your Scantron (shown in the figure below), bubble:
  1. Your Name
  2. Your NetID
  3. Form letter "A"
  4. Bubble the corresponding 3-digit code (shown below) for your lab section on your Scantron.

<table>
<thead>
<tr>
<th>Time</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00-10:50</td>
<td>101</td>
<td>104</td>
<td>107</td>
<td>109</td>
</tr>
<tr>
<td>11:00-12:50</td>
<td>102</td>
<td></td>
<td>110</td>
<td></td>
</tr>
<tr>
<td>1:00-2:50</td>
<td></td>
<td>105</td>
<td></td>
<td>111</td>
</tr>
<tr>
<td>3:00-4:50</td>
<td>103</td>
<td>106</td>
<td>108</td>
<td>112</td>
</tr>
</tbody>
</table>

- No electronic devices, books, notes, or cheat sheets are allowed while taking this exam.
- Please fill in the most correct answer on the provided Scantron sheet.
- We will not answer any questions during the exam.
- Each question has only ONE correct answer.
- You must stop writing when time is called by the proctors.
- Hand in both these exam pages and the Scantron.
- DO NOT turn this page UNTIL the proctor instructs you to.
1. The following C program compiles and runs without errors. What is the output of this program?

```c
#include <stdio.h>

void Rotate(int, int *, int);

void main(void)
{
    int x = 1;
    int y = 2;
    int z = 3;

    Rotate(x, &y, z);
    printf("%d %d %d\n", x, y, z);
}

void Rotate(int x, int * y, int z)
{
    int temp = x;

    x = *y;
    *y = z;
    z = temp;
}

(a) 1 2 3
(b) 3 2 1
(c) 2 3 1
(d) 1 3 3
```
2. Given the C program below, how many distinct variables exist in the program?

```c
#include <stdio.h>

int x = 10;
int y = 20;
int z = 30;

void add(int x, int y)
{
    z = x + y;
    printf("%i", z);
}

void sub(void)
{
    int x = 5;
    int y = 1;
    z = x - y;
    printf("%i", z);
}

void mult(int x, int y)
{
    z = x * y;
    printf("%i", z);
}

void main(void)
{
    int z = 1;
    printf("%i", z);
    add(x, y);
    sub();
    x = 1;
    y = 2;
    mult(x, y);
    printf("%i", z);
}
```

(a) 3  
(b) 5  
(c) 6  
(d) 10
3. Fill in the blanks to complete a comparison function named `comp` used by the function `qsort` to sort an array of integers in **DESCENDING** order.

```c
qsort(array, elts, sizeof(int), comp);

int comp(int *ptr1, int *ptr2)
{
    if(*ptr1 > *ptr2)
        return ___(i)___;
    else if(*ptr1 < *ptr2)
        return ___(ii)___;
    else return ___(iii)___;
}

(a) (i) 1 (ii) -1 (iii) 0
(b) (i) -1 (ii) 1 (iii) 0
(c) (i) 0 (ii) -1 (iii) 1
(d) (i) 0 (ii) 1 (iii) -1
```
4. Fill the blank to sort the values and only the values the user enters at the keyboard. The code for the function `compAsc` is not shown.

```c
#include <stdio.h>
#include <stdlib.h>
#define MAX_POSSIBLE_ELEM 10
#define ELEM_IN_ARRAY 5

int compAsc(int *, int *);

void main(void)
{
    int array[MAX_POSSIBLE_ELEM];
    int i;

    for(i = 0; i < ELEM_IN_ARRAY; i ++)
        scanf("%i", &array[i]);

    qsort(array, _______________ , sizeof(array[0]), compAsc);

    /* more code not shown */
}

(a) int
(b) MAX_POSSIBLE_ELEM
(c) ELEM_IN_ARRAY
(d) int *
5. The following program compiles and runs without errors. What is the output?

```c
#include <stdio.h>
int x = 1;  /* global variable */
int myFunction(int); /* function prototype */

void main(void)
{
    int result = 2;
    result = myFunction(result);
    printf("%i %i", result, x);
}

int myFunction(int x)
{
    printf("%i ", x);
    return x + 2;
}
```

(a) 2 4 1
(b) 2 4 2
(c) 2 2 1
(d) 2 2 2
6. What is the output of the following function call with arguments mentioned below?

```c
/* Function call */
int arr[] = {1, 2, 4, 6, 7};
printf("%i", Search(arr, 0, 4, 6));

/* Function definition */
int Search(int array[], int first, int last, int value)
{
    int midpt;
    if(first > last)
        return -1;
    else
    {
        midpt = (first + last)/2;
        if(value < array[midpt])
            return Search(array, first, midpt - 1, value);
        else if(value > array[midpt])
            return Search(array, midpt + 1, last, value);
        else
            return midpt + 1;
    }
}
```

(a) 6
(b) 7
(c) 3
(d) 4

7. Given the following pointer declarations, which of the following C statements is NOT a valid way of allocating an array for 100 integer values?

```c
int *ptr;

(a) ptr = calloc(100, sizeof(int));
(b) ptr = calloc(100, sizeof(ptr[0]));
(c) ptr = calloc(100, sizeof(ptr*));
(d) ptr = calloc(sizeof(int), 100);
```
8. What action does fopen take below when mode w is selected assuming that the file named fileName already exists?

FILE * ptr = fopen(fileName,"w");

(a) Returns the NULL value
(b) Opens fileName and writes at the end of the file
(c) Opens fileName for reading
(d) Opens fileName and deletes any existing content and writes in the empty file

9. You are given the following structure definition.

typedef struct
{
    double x;
    double y;
} vector;

Complete the function named norm that computes and returns the norm of a vector. The input to norm is a pointer to a vector. Recall that the norm of a vector with values (a,b) is \( \sqrt{a^2 + b^2} \). You can use functions from the C math library.

```c
double norm(vector * v)
{
    _______________________  
}
```

Choose the answer that by filling in the blank above gives the correct output.

(a) return sqrt((v.x)*(v.x) + (v.y)*(v.y));

(b) return sqrt((v->x)*(v->x) + (v->y)*(v->y));

(c) return sqrt((*v.x)*(*v.x) + (*v.y)*(*v.y));

(d) return sqrt((vector->x)*(vector->x) + (vector->y)*(vector->y));
10. Which of the following C code fragments would enable us to define a new data type called book, a structure data type so that we could declare the following array without errors?

book books[1000];

(a) typedef
    {
        char title[58];
        char author[33];
        int number_of_pages;
        float price;
    } book;

(b) typedef struct
    {
        char title[58];
        char author[33];
        int number_of_pages;
        float price;
    } books;

(c) typedef struct
    {
        char title[58];
        char author[33];
        int number_of_pages;
        float price;
    } book;

(d) typedef
    {
        char title[58];
        char author[33];
        int number_of_pages;
        float price;
    } books[1000];
11. The following program compiles and runs without any errors. Write the output of this program.

```c
#include <stdio.h>
#include <string.h>

typedef struct {
    int id;
    char name[100];
} Employee;

Employee func1(Employee emp) {
    emp.id = emp.id + 1000;
    strcpy(emp.name,"Steve Jones");
    return emp;
}

void func2(Employee * emp2) {
    if(strstr(*(emp2).name,"Jones") != NULL) {
        emp2->id = emp2->id - 2000;
    } else {
        emp2->id = emp2->id - 1000;
    }
}

void main(void) {
    Employee emp1 = {5525, "Mark Jones"};
    Employee emp2;
    emp2 = func1(emp1);
    printf("%i, %s\n", emp2.id, emp2.name);
    func2(&emp1);
    printf("%i, %s\n", emp1.id, emp1.name);
}

(a) 7525, Steve Jones
     3525, Mark Jones

(b) 6525, Steve Jones
     3525, Mark Jones

(c) 6525, Steve Jones
     4525, Mark Jones

(d) 3525, Steve Jones
     1525, Mark Jones
12. Assume the data type complex has been defined as shown below.

typedef struct
{
    double real;
    double imaginary;
} complex;

Assume that the variables w, ptr and z have been declared as follows:
complex w ={3.0, 4.0};
complex * ptr = &w;
double z;

Pick the statement that is NOT valid (a statement is not valid if it produces a compiler error).

(a) z = w->real;

(b) z = ptr->real;

(c) z = w.real;

(d) z = (*ptr).real;
13. The following program compiles and runs without errors. What is the output of the program?

```c
#include <stdio.h>

typedef struct{
    int x;
    int y;
    int z;
} Location;

void main(void)
{
    Location coordinates[10];
    Location * ptr = &coordinates[2]; /* ptr points to an element of the coordinates array */

    ptr->x = 11;
    (*ptr).y = 12;
    (*ptr).z = 1;

    coordinates[0].x = 1;
    coordinates[0].y = 12;
    coordinates[0].z = 11;

    printf("(%d, %d, %d)\n(%d, %d, %d)\n",
        ptr->x,
        ptr->y,
        ptr->z,
        coordinates[0].x,
        coordinates[0].y,
        coordinates[0].z);
}

(a) {11, 12, 1}
    {1, 12, 11}

(b) {11, 12, 11}
    {11, 12, 1}

(c) {1, 12, 11}
    {1, 12, 11}

(d) {12, 11, 1}
    {12, 11, 1}
```
14. In the movie *Inception*, the main character Cobb dreams in his dreams in his dreams in his dreams...
This resembles ‘recursion’. The following C program compiles and runs without errors. Write the output produced by the program.

```c
#include <stdio.h>

void dream(int n)
{
    if(n == 1)
    {
        printf("work ");
        return;
    }
    if(n > 1)
    {
        printf("love ");
        dream(n - 1);
        printf("play ");
    }
}

void main(void)
{
    dream(3);
}

(a) love work love play play
(b) work play love work play
(c) work work love play play
(d) love love work play play
```
The following program compiles and runs without errors. What is the output of this program?

```c
#include <stdio.h>

int f(int n)
{
    if(n < 2)
        return n;
    else
        return f(n - 1) + f(n - 2);
}

void main(void)
{
    int i;
    for(i = 0; i < 4; ++i)
        printf("%i ", f(i));
    printf("\n");
}

(a) 0 2 1 1
(b) 0 1 1 2
(c) 1 0 2 2
(d) 0 1 2 2
```
Use the following data type named `Student` and array named `students` to answer the next two questions.

```
typedef struct{
    char name[32];
    float studentno;
} Student;
```

```
Student students[10];
```

16. What is the correct syntax for declaring a one-dimensional 10-element array named `rollcall` which has elements of data type `Student`?

(a) `Student rollcall[10];`

(b) `Student rollcall[9];`

(c) `struct Student rollcall[9];`

(d) `Student struct[10] rollcall;`

17. Which of the following is a valid C statement that will assign the name "Michael Jordan" to the `name` field of the first element of the array named `students`?

(a) `students[0].name = "Michael Jordan";`

(b) `(*students).name = "Michael Jordan";`

(c) `strcpy(students – > name, "Michael Jordan");`

(d) `strcpy(name, "Michael Jordan");`
18. The following program compiles and runs without errors. What is the output of the program?

```c
#include <stdio.h>
#include <string.h>

typedef struct {
    char name[32];
    int age;
} animal;

void main(void)
{
    animal pet = {"dog", 3};
    char * ptrA = "cat";
    char * ptrB;
    printf("%s ", pet.name);
    printf("%c ", pet.name[1]);
    strcpy(pet.name, ptrA);
    ptrB = pet.name;
    printf("%c ", ptrB[0]);
    printf("%c ", *ptrB);
}

(a) dog o a t
(b) cat c a d
(c) dog o c c
(d) cat a d d
The program below compiles and executes without errors. What is the output of this program?

```c
#include <stdio.h>

typedef struct
{
    int a[10];
    int b[10];
} pair;

void eleven(int a[], int b[])
{
    int j;

    for(j = 0; j < 10; j++)
        a[j] = 11 % b[j];
}

void main(void)
{
    int i;
    pair z;

    for(i = 0; i < 10; i++)
        z.a[i] = i + 1;
    eleven(z.b, z.a);
    printf("%i\n", z.a[3]);
}

(a) 3
(b) 4
(c) 6
(d) 5
```
The assigned values for \( S \) and \( \text{ptrS} \) are given below. Given that the address in memory of \( S \) is 3000 and the address of \( \text{ptrS} \) is 3004 answer the following two questions.

```c
int S = 7;
int * ptrS = &S;
*ptrS = 4;
```

20. After the code above is executed what value does \( S \) hold?

(a) 7

(b) 3000

(c) 3004

(d) 4

21. After the code above is executed what value does \( \text{ptrS} \) hold?

(a) 7

(b) 3000

(c) 3004

(d) 4

22. What are the values of the array named \( P \) after the following C code is executed?

```c
int P[] = {1, 2, 3};
int * ptrP = P;
*ptrP = 4;
```

(a) \{4, 2, 3\}

(b) \{1, 2, 3\}

(c) \{1, 2, 4\}

(d) \{4, 4, 4\}
23. Given the following pointer declaration,

```
int * ptr;
```

which of the following is a valid line of C code?

(a) `ptr = &malloc(100, sizeof(int));`

(b) `ptr = malloc(400);`

(c) `ptr == malloc();`

(d) `ptr = *malloc(100, sizeof(int));`
The following incomplete code is supposed to sort elements of an array of data type Student according to values in the field named uin in **ASCENDING** order. The next two questions will ask you to complete the code.

```c
#include <stdio.h>
#include <stdlib.h>

typedef struct
{
    int uin;
    char name[20];
} Student;

int compuinAsc(Student *p1, Student *p2)
{
    /* code for compuinAsc goes here */
}

void main(void)
{
    Student cs101[10];
    int i;
    for(i = 0; i < 10; i++)
    {
        scanf("%i", &cs101[i].uin);
        scanf("%s", cs101[i].name);
    }
    /* call qsort here */
    /* more code not shown here */
}
```

24. Which code completes the function named compuinAsc used above?

(a) if(p1− > uin > p2− > uin)
    return -1;
else if(p1− > uin == p2− > uin)
    return 0;
return 1;

(b) return (*p2).uin − (*p1).uin;

(c) return p1− > uin − p2− > uin;

(d) return *p1.uin − *p2.uin;
25. Which of the following C statements correctly calls the `qsort` function to sort the array named `cs101` in **ASCENDING** order by the value in the `uin` field?

(a) `qsort(cs101.uin, 10, sizeof(int), compuinAsc);`

(b) `qsort(cs101.uin, 10, sizeof(Student), compuinAsc);`

(c) `qsort(cs101, 10, sizeof(Student), compuinAsc);`

(d) `qsort(cs101, 10, sizeof(int), compuinAsc);`
Extra Credit

26. The following program compiles and runs without errors. What is the output of the program?

```c
#include <stdio.h>

typedef struct{
    double real;
    double imaginary;
} complex;

complex add(complex, complex); /* prototype */

void main(void)
{
    complex x = {2.0, 4.0}, y, z;
    z = x;
    y = add(x, z);
    printf("real(y) = %lf, imag(y) = %lf\n", y.real, y.imaginary);
}
complex add(complex a, complex b)
{
    complex c = {a.real + b.real, a.imaginary + b.imaginary};
    return c;
}

(a) real(y) = 2.000000, imag(y) = 4.000000
(b) real(y) = 4.000000, imag(y) = 4.000000
(c) real(y) = 4.000000, imag(y) = 8.000000
(d) real(y) = 8.000000, imag(y) = 8.000000
```
27. The following program compiles and runs without errors. What is the output of the program?

```c
#include <stdio.h>

void main(void)
{
    char str[] = "crown";
    char * ptrA = "royal";
    char * ptrB = str;

    printf("\"%s \"", str);
    printf("\"%s \"", ptrA);
    printf("\"%s \"", ptrB);
    printf("\%c %\c", ptrA[2], *ptrB);
}
```

(a) crown royal crown y c

(b) crown royal crown yal c

(c) crown royal crown yal crown

(d) royal crown royal y c