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.
STOP!!! Please ensure that your NETID is bubbled in correctly on your scantron sheet. If it is not, you will AUTOMATICALLY be deducted 5 points from your final score.
1. 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)
{
    int 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) 10  
(d) 11
2. The following C program compiles and runs without errors. What is the output produced by this program?

```c
#include <stdio.h>

int x = 5;

void func(int * x)
{
    *x = 6;
}

void main(void)
{
    int * p = &x;
    func(p);
    printf("%i, %i\n", x, *p);
}
(a) 5, 5
(b) 5, 6
(c) 6, 5
(d) 6, 6
```

3. The following C program compiles and runs without errors. What is the output produced by this program?

```c
#include <stdio.h>

void main(void)
{
    int a[] = {10, 20, 30, 40, 50};
    int *c;
    int *b = a;

    printf("%i ", *(b++));
    c = b;
    printf("%i ", ++(*c));
}
(a) 10 21
(b) 20 21
(c) 10 20
(d) 11 20
```
4. The following C program compiles and runs without errors. What is the output produced by this program?

```c
#include <stdio.h>

void main(void)
{
    /* there is a blank space before the H in " Hello" */
    char str[20] = " Hello";
    char *p = str;

    *p = 'M';
    *(p+3) = '\0';
    printf("%s\n", str);
}
```

(a) MHe  
(b) Hello  
(c) Mello  
(d) MHello

5. Replace the blank line below with the correct line of C code in order to output an array containing the individual ASCII values of the letters that were input. The final program should not produce any errors when compiled or executed.

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

void main(void)
{
    int *ascii,i;
    char word[50];

    printf("Enter word: ");
    scanf(" %[^\n]",word);

    _______________________________________
    printf("The ASCII values are: ");
    for (i=0; i<strlen(word); ++i)
    {
        ascii[i] = word[i];
        printf("%i ",ascii[i]);
    }
    free(ascii);
}
```

(a) word = calloc(strlen(word) * sizeof(int));  
(b) word = malloc(strlen(word), sizeof(int));  
(c) ascii = calloc(strlen(word), sizeof(int));  
(d) ascii = malloc(strlen(word), sizeof(char));
6. The following C program compiles and runs without errors. What is the output produced by this program?

```c
#include <stdio.h>

void func1(int x, int a[])
{
    a[x] = a[x] + 1;
}

void func2(int y, int * b)
{
    b[y] = b[y] + 1;
}

void main(void)
{
    int c[] = {10, 20, 30};
    int d[] = {10, 20, 30};

    func1(0, &c[0]);
    printf("%i %i %i \n", c[0], c[1], c[2]);

    func2(1, &d[1]);
    printf("%i %i %i \n", d[0], d[1], d[2]);
}
```

(a) 10 20 30
    10 21 30

(b) 11 20 30
    10 20 31

(c) 10 21 30
    10 21 30

(d) 10 21 30
    10 20 31
7. Replace the blank line below with the correct line of C code in order to get the output 10. The final program should not produce any errors when compiled or executed.

```c
#include <stdio.h>

void main(void)
{
    int a=5, b=10, c=15;
    int *arr[]={&a, &b, &c};
    _________________
}
```

(a) printf("%i \n", arr[1]);
(b) printf("%i \n", *arr[1]);
(c) printf("%i \n", arr[0]);
(d) printf("%i \n", *arr[0]);

8. The following C program compiles and runs without errors. What is the output produced by this program?

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

int comp(int *ptr1, int *ptr2)
{
    return *ptr2 - *ptr1;
}

void main(void)
{
    int i;
    int Array[7] = {3, 10, 1, 5, 10, 7, 11};

    qsort(Array, 7, sizeof(Array[0]), comp);

    for(i=0; i < 7; ++i)
        printf("%i ", Array[i]);
}
```

(a) 1 3 5 7 10 10 11
(b) 1 3 5 7 10 11
(c) 11 10 10 7 5 3 1
(d) 11 10 7 5 3 1
9. Which answer choice, containing the code for the `compare` function and call to the `qsort` function, when replacing the corresponding blank lines in the program shown below, would correctly sort the array of words declared below in **ascending** alphabetical order?

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

typedef char string[32];

_____ code for compare function goes here__________

void main(void)
{
    int i;
    string words[7] = {"this", "that", "those", "these", "them", "abc", "abcd"};

    __________call to qsort goes here__________________

    for(i = 0; i < 7; ++i)
    {
        printf("%s ", words[i]);
    }

(a) int compare( char * ptr1, char * ptr2)
{
    return strcmp( ptr1, ptr2);
}

qsort(words, 7, sizeof(string), compare);

(b) int compare( char * ptr1, char * ptr2)
{
    return strcmp( ptr1, ptr2);
}

qsort(words, 7, sizeof(char), compare);

(c) int compare( char * ptr1, char * ptr2)
{
    return *ptr1 - *ptr2;
}

qsort(words, 7, sizeof(string), compare);

(d) int compare( char * ptr1, char * ptr2)
{
    return *ptr1 - *ptr2;
}

qsort(words, 7, sizeof(char), compare);
```
10. Given the following C code fragment, which of the following statements is \textbf{TRUE}?

```c
typedef struct {
    int id;
    char name[32];
    int quantity;
    float price;
} part;
```

(a) \texttt{part} is the name of a variable and the code above allocates memory for each field;
(b) \texttt{part} is the name of a structure and the code above allocates memory for each field;
(c) \texttt{part} is the name of a variable and the code above does not allocate memory for each field;
(d) \texttt{part} is the name of a structure and the code above does not allocate memory for each field;

11. Which of the following statements is \textbf{TRUE} about \textit{structures}? Hint: The following answer choices refer to an argument having a structure data type, which does \textbf{NOT} include an array of structure data type.

(a) Function calls with an argument having a structure data type are implemented as \textbf{call-by-value}, and functions \textbf{can} return values of structure data type.
(b) Function calls with an argument having a structure data type are implemented as \textbf{call-by-reference}, and functions \textbf{can} return values of structure data type.
(c) Function calls with an argument having a structure data type are implemented as \textbf{call-by-value}, and functions \textbf{cannot} return values of structure data type.
(d) Function calls with an argument having a structure data type are implemented as \textbf{call-by-reference}, and functions \textbf{cannot} return values of structure data type.

12. Which of the following C statements can be used to declare two pointer variables? Both variables are pointers to \texttt{float} data type.

(a) \texttt{float *fptr1, fptr2;}
(b) \texttt{float *fptr1, *fptr2;}
(c) \texttt{float *(fptr1, fptr2);} 
(d) None of the above.
13. Replace the blank line with the correct answer choice to sort the values and ONLY the values that 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 *); /* prototype */

int 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 *
14. Complete the following C program, by replacing the blank line with the correct answer choice, to read in ten albums of information in the format described below. We want to write a program to read in ten CD albums of information.

Each CD album data has the following format:
artist name; CD title; number of tracks(songs)

and one specific example is the following:
Time Out; The Dave Brubeck Quartet; 7

Further each track has the format:
track name; minutes; seconds

with a specific example like:
Take Five; 5; 24

```c
#include <stdio.h>

typedef struct{
    int min;
    int sec;
} Time;

typedef struct{
    char name[32];
    Time time;
} Track;

typedef struct {
    char title [125];
    char artist[100];
    int num_tracks;
    Track tracks[20];
} Album;

void main(void) {
    int i, j;
    Album albums[10];

    for(i = 0; i < 10; ++i)
    {
        scanf(" %[^;]; %[^;]; %i",albums[i].artist,albums[i].title,&albums[i].num_tracks);
        for(j = 0; j < albums[i].num_tracks; ++j)
        { } // more code not shown here */
    }

    (a) scanf(" %[^;]; %i; %i", albums[j].name, &albums[j].time.min, &albums[j].time.sec);
    (b) scanf(" %[^;]; %i; %i", albums[i].tracks[j].name, albums[i].tracks[j].time.min, albums[i].tracks[j].time.sec);
    (c) scanf(" %[^;]; %i; %i", albums[i].tracks[j].name, &albums[i].tracks[j].time.min, &albums[i].tracks[j].time.sec);
    (d) scanf(" %[^;]; %i; %i", albums[j].name, albums[j].time.min, albums[j].time.sec);
```
You are given the following structure definition for the **NEXT THREE** questions.

```c
typedef struct {
    double length;
    double height;
    double width;
} object;
```

15. Complete the function named `calc_volume` by selecting the correct answer choice, which when replacing the blank line below, computes and returns the volume of a rectangular box of dimensions length, height and width. The input to `calc_volume` is a pointer to an `object`. Recall that the volume of a rectangular box is length*height*width.

```c
double calc_volume(object *o)
{
    ______________________________
}
```

(a) `return (o.length)*( o.height)*( o.width);`
(b) `return (o->length)*( o->height)*( o->width);`
(c) `return (*o.length) * (*o.height) * (*o.width);`
(d) `return (object->length)*(object->height)*(object->width);`

16. You are given the array named `Obj` declared below. Which of the following choices will assign the value 5 to the width field of the tenth element of the array named `Obj`?

```c
object Obj[50];
```

(a) `strcpy(Obj[9].width , 5);`
(b) `&(Obj + 9).width = 5;`
(c) `Obj[9].width = 5;`
(d) `Obj[9].width.5;`

17. Suppose that the variables `o`, `ptr` and `ans` have been declared as shown below.

```c
object o = {3.0, 4.0, 5.0};
object * ptr = &o;
double ans;
```

Which of the following statements is **NOT** a valid C statement (it will produce a compiler error)?

(a) `ans = o -> length;`
(b) `ans = ptr -> length;`
(c) `ans = o.length;`
(d) `ans = (*ptr).length;`
18. Given the structure data type named movies defined below, which is the correct way to declare a variable named m of data type movies and assign the first day revenue of the 2012 James Bond action movie Skyfall with the value of $250,000,000? (Skyfall is the title and action is the genre)

```c
typedef struct {
    double revenue;
    char title[200];
    int year;
    char genre[50];
} movies;
```

(a) movies m = {"Skyfall", 2012, 250000000, "action"};
(b) movies m = [250,000,000, "Skyfall", 2012, "action"];
(c) movies m = {250000000, "Skyfall", 2012, "action"};
(d) movies {250000000, 2012, "action", "Skyfall"};

19. Which of the following statements about fopen() is NOT True?

(a) When the mode "r" is used, and the specified file does not exist, fopen() returns NULL.
(b) When the mode "w" is used, and the specified file does not exist, fopen() returns NULL.
(c) When the mode "a" is used, and the specified file does not exist, fopen() creates a file.
(d) When the mode "a" is used, and the specified file exists, fopen() opens the file for writing at end of the file.
20. Consider the following C code below. The program should sum all the integers from a file named infile.dat but the program does NOT work correctly. The line numbers are not part of the code.

```c
1. #include <stdio.h>
2. void main(void)
3. {
4.     FILE *fileIn;
5.     int value;
6.     int total;
7.     fileIn = fopen("infile.dat", "r");
8.     while (fscanf(fileIn, "%i", value))
9.         total += value;
10.    printf("total = %i \n", total);
11.    fclose(fileIn);
12. }
```

Which of the following statements is NOT True?

(a) Since fopen() may return NULL at line 8, proper error handling code is missing from this program.

(b) The logical condition at line 10 does not work correctly if end of file (EOF) is encountered.

(c) fclose(fileIn); must be replaced with free(fileIn); in order for the program to work correctly.

(d) At line 10, value should be replaced by &value in order for the program to work correctly.

21. Which of the following C statements will correctly open a file with the name infile.dat for appending.

(a) FILE *fin;
    fin = fopen("infile.dat","a");

(b) FILE *fin;
    *fin = fopen("infile.dat","r");

(c) FILE *fin;
    fin = fopen("infile.dat","w");

(d) FILE *fin;
    fin = fopen("infile.dat","a");
The following program compiles and runs without errors. What is the output that it produces?

```c
#include <stdio.h>

int seq_search(int [], int, int, int); /* Prototype */

void main(void)
{
    int list[] = {1,10,4,3,2,7,6,9,34,21,44,67,5};
    int lb = 5;
    int ub = 10;
    int item = 9;
    int index = -1;

    index = seq_search(list, lb, ub, item);

    if(index != -1)
        printf("%i", index);
    else
        printf("Not found!");
}

int seq_search(int arr[], int i, int j, int item)
{
    if(i<=j)
    {
        if(arr[i] == item)
            return i;
        else
            return seq_search(arr, i+1, j, item);
    }
    else
        return -1;
}

(a) 9
(b) 7
(c) 2
(d) 3
23. **TRUE** or **FALSE**? A recursive function can call itself.

(a) TRUE

(b) FALSE

24. The following program will compile and run without errors. What is the output that it produces?

```c
#include <stdio.h>

int euclidean(int, int); /* Prototype */

void main(void)
{
    int num1 = 100;
    int num2 = 35;
    printf("%i", euclidean(num1, num2));
}

int euclidean(int num1, int num2)
{
    if(num1%num2 == 0) /* VOOM!! */
        return num2;
    else
        return euclidean(num2, num1%num2);
}

(a) 30

(b) 0

(c) 5

(d) 35
25. Which of the answer choices below, when inserted in place of the blank line in the code below, will complete the function \texttt{babylonian} so that it correctly implements the recursive Babylonian square root algorithm outlined below?

The Babylonian algorithm is as follows:

1. Start with a guess of the square root of \( N \), call it \( x_0 \)
2. The next approximation of the square root of \( N \) is \( x_{n+1} \) and is the arithmetic mean of \( x_n \) and \( \frac{N}{x_n} \)
3. Repeat step 2 until a certain desired accuracy is achieved

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

double babylonian(double N, double x, double error)
{
    if(fabs(N-x*x) < error) /* base case (VOOM!!) */
        return x;
    else
    {
        /* calculate the new value for the square root of N based */
        /* on the Babylonian algorithm */
        x = 0.5*(x + N/x);
        /* calculate the new value for the square root of N based */
        /* on the Babylonian algorithm */
        x = 0.5*(x + N/x);
    }
}

void main(void)
{
    double error = 1.0e-6;
    double N = 9.00;
    double x0 = 2.00; /* guess */
    double answer;

    answer = babylonian(N, x0, error);

    /* print the answer */
    printf("\n approximation to square root of %lf: %.15lf \n", N, answer);
}

(a) return x;
(b) return 0.5*(x + N/x);
(c) return babylonian(N+1, x, error);
(d) return babylonian(N, x, error);
```
Extra Credit

Answering, the next two questions below correctly will add points to your exam total. Answering incorrectly or not answering will not add points to your exam total.

26. Which of the following is the correct way to write a comparison function named \texttt{compOddEven}? The function \texttt{compOddEven} sorts the odd numbers before the even numbers. For example, when used in a call to \texttt{qsort} in the following C program, the program prints the values 3 1 2 4.

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

int compOddEven(int *, int *);

void main(void)
{
    int array[4] = {2,3,4,1};
    int i;

    qsort(array, 4, sizeof(int), compOddEven);

    for(i = 0 ; i < 4; ++i)
        printf("%i ", array[i]);
}

(a) int compOddEven(int * ptrL , int * ptrR) {
    if( ( (*ptrR)%2 == 0) && ((*ptrL)%2 == 1) )
        return 1;
    else
        return -1;
}

(b) int compOddEven(int * ptrL , int * ptrR) {
    if( ( (*ptrL)%2 == 0) && ((*ptrR)%2 == 1) )
        return 1;
    else
        return -1;
}

(c) int compOddEven(int * ptrL , int * ptrR){
    if( ( (*ptrL)%2 == 0) || ((*ptrR)%2 == 1) )
        return 1;
    else
        return -1;
}

(d) int compOddEven(int * ptrL , int * ptrR){
    if( ( (*ptrR)%2 == 0) || ((*ptrL)%2 == 1) )
        return 1;
    else
        return -1;
}
The following C program compiles and runs without errors. What is the output produced by this program?

```c
#include <stdio.h>

void main(void)
{
    int x = 3;
    int y = 4;
    int *p = &x;
    int * *q = &p;

    *p = 5;
    p = &y;
    *(q) = 6;

    printf("x = %i  y = %i\n", x,y);
}
```

(a) x = 6  y = 4
(b) x = 6  y = 5
(c) x = 4  y = 5
(d) x = 5  y = 6