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>108</td>
<td>111</td>
</tr>
<tr>
<td>11:00-12:50</td>
<td>102</td>
<td>105</td>
<td>109</td>
<td>112</td>
</tr>
<tr>
<td>1:00-2:50</td>
<td></td>
<td>106</td>
<td></td>
<td>113</td>
</tr>
<tr>
<td>3:00-4:50</td>
<td>103</td>
<td>107</td>
<td>110</td>
<td>114</td>
</tr>
</tbody>
</table>

Five points will be deducted if a student does not correctly record their netid on the bubble sheet.

- 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. **No extra time will be given after the exam ends to fill in bubble sheets with answers.**
- Hand in both these exam pages and the Scantron.
- DO NOT turn this page UNTIL the proctor instructs you to.
1. Given the C program below, how many distinct variables exist in the program?

```c
#include <stdio.h>

int x = 0;
int i = 0;

int function1(int x, int y)
{
    return x-y;
}

void function2(int z)
{
    printf("%i",z);
}

void main(void)
{
    int y = 1;
    int s = function1(x,y);
    function2(s);
}
```

(a) 2
(b) 5
(c) 6
(d) 7
2. The following C programs compile and run without errors. Which one has a different output from the others?

(a) #include <stdio.h>

    int x = 3;

    void main(void)
    {
        printf("%i", x);
    }

(b) #include <stdio.h>

    int x = 0;

    void main(void)
    {
        int x = 3;
        printf("%i", x);
    }

(c) #include <stdio.h>

    int x = 3;

    void function(int *y)
    {
        *y = *y + 1;
    }

    void main(void)
    {
        function(&x);
        printf("%i", x);
    }

(d) #include <stdio.h>

    int x = 3;

    void function(int x)
    {
        int *xPtr = &x;
        *xPtr = *xPtr + 1;
    }

    void main(void)
    {
        function(x);
        printf("%i", x);
    }
3. In the following code, how many distinct pointer variables have been declared?

```c
int x, y;
int *ptr1, ptr2;
```

(a) 1  
(b) 2  
(c) 3  
(d) 4

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

```c
#include <stdio.h>

void main(void)
{
    int y;
    int *ptr1;
    y = 3;
    printf("y = %i\n", y);
    ptr1 = &y;
    *ptr1 = 2;
    printf("y = %i\n", y);
}
```

(a) y = 2
    y = 3  
(b) y = 3
    y = 2  
(c) y = 3
    y = 3  
(d) y = 2
    y = 2

5. Consider the following variable declarations:

```c
int a = 5;
int *c = &a;
```

Which of the following statements will produce a compile error when typed in a complete C program that includes the above declarations?

(a) c = *a;  
(b) a = *c;  
(c) a = 2;  
(d) *c = 3;
6. Assume the following lines of code appear in a C program.

```c
int x = 1;
int y = 2;
int * ptr1;
int * ptr2;

ptr1 = &x;
ptr2 = &y;
*ptr1 = 3;
*ptr2 = 4 + *ptr1;
```

When the code above is placed in a complete C program that compiles and runs without errors, the operating system stores the variables at memory addresses shown in the table below. Fill in the following table with the values stored in memory after the code above executes.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>(i) 1</td>
<td>4000</td>
</tr>
<tr>
<td>y</td>
<td>(ii) 2</td>
<td>4004</td>
</tr>
<tr>
<td>ptr1</td>
<td>(iii) 3</td>
<td>4008</td>
</tr>
<tr>
<td>ptr2</td>
<td>(iv) 7</td>
<td>4016</td>
</tr>
</tbody>
</table>

(a) (i) 1
   (ii) 2
   (iii) 3
   (iv) 7

(b) (i) 1
    (ii) 2
    (iii) 3
    (iv) 5

(c) (i) 1
    (ii) 2
    (iii) 4000
    (iv) 4004

(d) (i) 3
    (ii) 7
    (iii) 4000
    (iv) 4004
7. 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 array[5] = {0, 10, 20, 30, 40};
    int *ptr_array[5];

    int i;
    for(i = 0; i < 5; i++)
    {
        ptr_array[i] = &array[i];
    }

    printf("%i", *(++ptr_array[0]));
}
```

(a) 0
(b) 1
(c) 10
(d) 11

8. Complete the program below by choosing the correct code fragment, such that when the blank in the program is replaced with the code fragment, the following values will print when the program is compiled and run.

3 5 7

```c
#include <stdio.h>

void main(void)
{
    int a = 3;
    int b[2] = {7, 5};

    int *x = &a;
    int *y = &b[0];

    printf("%i %i %i", __________);
}
```

(a) *x, y[1], *y
(b) &x, y, *y
(c) *x, *y[1], y
(d) &x, *y, y
9. The following program compiles and runs without errors. What is the output of this program?

```c
#include <stdio.h>

void swap(int *a, int *b)
{
    int tmp = *a;
    *a = *b;
    *b = tmp;
}

void swap3(int *a, int *b, int *c)
{
    int tmp = *a;
    *a = *b;
    *b = *c;
    *c = tmp;
}

void main(void)
{
    int a[2]={ 1, 0};
    int b[2]={ 3, 2};

    swap( &a[0], &a[1] );
    printf("%i %i\n", a[0], a[1] );

    swap3( &b[0], &b[1], &b[0] );
    printf("%i %i\n", b[0], b[1] );
}
```

(a) 0 1
    2 3
(b) 0 1
    3 2
(c) 1 0
    2 3
(d) 1 0
    3 2
10. Which of the code fragments below is a correct way to use Dynamic Memory Allocation to create an array named `a` with 300 elements of data type `int`?

- i. `int *a; a = malloc(300*sizeof(int));`
- ii. `int a; a = malloc(300, sizeof(int));`
- iii. `int a; a = calloc(300*sizeof(int));`
- iv. `int *a; a = calloc(300, sizeof(int));`

(a) ii and iii  
(b) i and iii  
(c) i and iv  
(d) iii and iv  

11. The following C program compiles and runs without errors. What is the output of this program?

```c
#include <stdio.h>  
#include <stdlib.h>  
#define ARRAY_SIZE 5

int comp(int * ptrL , int * ptrR)
{
    return -(*ptrR - *ptrL);
}

void main(void)
{
    int i;
    int nums[ARRAY_SIZE] = {5,4,3,2,1};

    for(i=0; i < ARRAY_SIZE; ++i)
        printf("%i ",nums[i]);

    qsort(nums,ARRAY_SIZE,sizeof(int),comp);

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

(a) 5 4 3 2 1 1 2 3 4 5  
(b) 5 4 3 2 1 5 4 3 2 1  
(c) 1 2 3 4 5  
(d) 5 4 3 2 1
12. Fill in the two blanks to complete a comparison function named `compare_function` used by the function `qsort`. We want to sort an array named `nums` in **ASCENDING** order.

```c
#include <stdio.h>
#include <stdlib.h>
#define ARRAY_SIZE 5

int compare_function (int *L, int *R)
{
    if ___(i)___
        return 1;
    else if ___(ii)___
        return -1;
    else
        return 0;
}

void main(void)
{
    int i;
    float nums[ARRAY_SIZE] = {3.2, 5.6, 1.2, 5.6, 8.9};

    qsort(nums,ARRAY_SIZE,sizeof(nums[0]),compare_function);

    for(i=0; i < ARRAY_SIZE; ++i)
        printf("%.2f ",nums[i]);
}

(a) (i) (*R > *L)
    (ii)(*R < *L)
(b) (i) (*L < *R)
    (ii)(*R > *L)
(c) (i) (*L <= *R)
    (ii)(*L >= *R)
(d) (i) (*L > *R)
    (ii)(*L < *R)
13. Fill in the blank to complete the comparison function named `fnc` used by the function `qsort`. The goal is to correctly sort the array of words in reverse alphabetical order (words starting with the letter `z` before words starting with letter `a`).

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

typedef char string[50];

int fnc( char * ptr1, char * ptr2)
{
    return __________________________;
}

void main(void)
{
    int i;
    string words[9] = {"please", "ensure", "that", "your", "netid", "is", "bubbled", "in", "correctly"};

    qsort(words, 9, sizeof(string), fnc);

    for(i = 0; i < 9; ++i)
        printf("%s ", words[i]);
}

(a) strcmp(ptr1, ptr2)
(b) strcmp(ptr2, ptr1)
(c) (ptr2 - ptr1)
(d) -(ptr2 - ptr1)
```
14. The following C program compiles and runs without errors. What is the output of this program?

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

typedef struct
{
    int num; /* numerator */
    int den; /* denominator */
} Fraction; /* numerator / denominator */

int comp(Fraction * s, Fraction * t)
{
    return ((s -> num) % (s -> den)) - ((t -> num) % (t -> den));
}

void main(void)
{
    Fraction a[2];
    int i;
    a[0].num = 9;
    a[0].den = 2;
    a[1].num = 10;
    a[1].den = 5;
    qsort(a, 2, sizeof(a[0]), comp);
    for(i = 0; i < 2; i++)
        printf("%i/%i ", a[i].num, a[i].den);
}

(a) 10/5 9/2
(b) 5/10 9/2
(c) 9/2 10/5
(d) 1 0
The following program compiles and runs without errors. What is the output of this program?

```c
#include <stdio.h>

typedef struct
{
    float x;
    float y;
    float z;
    float time;
} Coords;

void swap1( Coords p )
{
    float temp = p.x;
    p.x = p.z;
    p.z = temp;
}

void swap2( Coords *q )
{
    float temp = q->x;
    q->x = q->z;
    q->z = temp;
}

void main( void )
{
    Coords r = {1.1, 2.2, 3.3, 10.0};

    swap1( r );
    printf("%.1f %.1f %.1f %.1f\n", r.x, r.y, r.z, r.time);
    swap2( &r );
    printf("%.1f %.1f %.1f\n", r.x, r.y, r.z);
}
```

(a) 3.3 2.2 1.1 10.0
    3.3 2.2 1.1
(b) 3.3 2.2 1.1 10.0
    1.1 2.2 3.3
(c) 1.1 2.2 3.3 10.0
    3.3 2.2 1.1
(d) 1.1 2.2 3.3 10.0
    1.1 2.2 3.3
The next three questions refer to the following two structure data types and array.

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

typedef struct
{
    char name[40];
    int age;
} Driver;

typedef struct
{
    char teamname[40];
    char type[40];
    int teampoints;
    Driver drivers[2];
} Team;

Team teams[50];

16. The program below (including the C code shown above) compiles and runs without errors. What is the output of this program?

```main(void)
{
    Driver d1 = {"Kimi Raikkonen", 33};
    Driver d2 = {"Danica Patrick", 31};
    Team t1 = {"Stewart-Haas Racing", "NASCAR", 120, {d2}};
    Team t2 = {"Lotus F1", "Formula1", 230, {d1}};
    printf("%s : %s : ", t2.type, t2.teamname);
    printf("%s : %i\n", t2.drivers[0].name, t2.drivers[0].age);
}
```(a) NASCAR : Stewart-Haas Racing : Danica Patrick : 31
(b) Formula1 : Lotus F1 : Danica Patrick : 31
(c) Formula1 : Lotus F1 : Kimi Raikkonen : 33
(d) NASCAR : Lotus F1 : Kimi Raikkonen : 33

17. What line of C code will assign 22 to the age field of the 1st driver (i.e., 1st element of the drivers array) in the 5th element of the array named teams declared above?

(a) teams[4].drivers[0].age = 22;
(b) teams[4].drivers[1].age = 22;
(c) teams[5].drivers[0].age = 22;
(d) 22 = teams[4].drivers[0].age;

18. Suppose we want to use qsort to sort the array named teams (declared above) in descending order by the value in the teampoints field. What should be the value returned by the following comparison function? The code should work correctly for any legal values assigned to the teams array.

```c
int compTeampointsDesc ( Team *L, Team *R )
{
    return ____________________________ ;
}
```(a) L->teampoints - R->teampoints
(b) L->teampoints + R->teampoints
(c) - ( R->teampoints - L->teampoints )
(d) R->teampoints - L->teampoints
#include <stdio.h>
#include <math.h>

typedef struct
{
    int size;     /* how many numbers are to be scanned? */
    int numbers[100];
} List;

/* isprime returns 1 if number is fermat prime else returns -1 */
int isprime (int * ptr)
{
    int n = 0;
    int value;
    do{
        value = pow(2,pow(2,n))+1;
        if (*ptr == value)
            return 1;
        ++n;
    }while(*ptr > value);
    return -1;
}

/* primes displays which numbers in the list are fermat primes */
void primes (List * f)
{
    int i = 0;
    int result;
    while (i < f->size)
    {
        result = isprime (&(f->numbers[i]));
        if (result != -1)
            printf(" %i is a fermat prime \n", f->numbers[i]);
        else
            printf(" %i is not a fermat prime \n", f->numbers[i]);
        i = i+1;
    }
}

void main(void)
{
    List list;
    int i;
    _____[1]_____   
    _____[2]_____
19. Which of the following code fragments, when placed at location [1] in the above C code, will read in the following data correctly when the user types in the data at the keyboard? (We are assuming that the entire program is correctly completed and compiles and runs without errors before the data is entered.)

The data is:

8, 3, 5, 17, 59, 0, 1, 33, 64,

(a) scanf("%i;", &list->size);
    for (i = 0; i < list->size; i++)
        scanf(" %i;", &list->numbers[i]);
(b) scanf("%i," list->size);
    for (i = 0; i < list->size; i++)
        scanf(" %i," list->numbers[i]);
(c) scanf("%i;", &list.size);
    for (i = 0; i < list.size; i++)
        scanf(" %i;", &list.numbers[i]);
(d) scanf("%i," &list.size);
    for (i = 0; i < list.size; i++)
        scanf(" %i," &list.numbers[i]);

20. In the above C code in the place of [2], which of the function calls shown below will call the function named primes with the correct input arguments? You may assume that [1] has been correctly coded.

(a) primes(list);
(b) primes(&list);
(c) primes(List * list);
(d) primes(*list);
21. Complete the program below by choosing the correct code fragments such that when the blanks in the program are replaced with the code fragments the program will read in three integers from the file named `text.dat`.

```c
#include <stdio.h>

void main(void)
{
    int a, b, c;
    FILE * fileptr;

    fileptr = fopen("text.dat", ___(i)______);

    fscanf(___ (ii)______, "%i %i %i", &a, &b, &c);

    printf("%i %i %i\n", a, b, c);
}

(a) (i) "w"
    (ii) FILE
(b) (i) "r"
    (ii) fileptr
(c) (i) "w"
    (ii) fileptr
(d) (i) "r"
    (ii) "text.dat"
```

22. Fill in the blank in the following program which reads in the data below typed by the user at the keyboard.

```
Favorite Numbers; 123.4; 12; 10.445
```

The output should appear as,
```
Favorite Numbers 123.40 12.00 10.45
```

```c
#include <stdio.h>

void main(void)
{
    char name[20];
    double a, b, c;

    scanf(______________, name, &a, &b, &c);

    printf("%20s %6.2lf %6.2lf %6.2lf\n", name, a, b, c);
}

(a) "%s; %lf; %lf; %lf"
(b) "%s; %lf; %lf; %lf;"
(c) "%[^;]; %lf; %lf; %lf"
(d) "%s %s; %lf; %lf; %lf"
```
23. The recursive program below compiles without errors. What is the output when you run the program?

```c
#include <stdio.h>

int dijk(int, int);

void main(void)
{
    int m = 6;
    int n = 4;
    int answer;

    answer = dijk(m, n);

    printf("%i \n", answer);
}

int dijk(int m, int n)
{
    if(m == n)
        return m; /* VOOM!! */
    else if(m > n)
        return dijk(m-n, n);
    else
        return dijk(m, n-m);
}

(a) 4
(b) 6
(c) the program goes into an infinite loop
(d) 2
The recursive program below compiles and runs without errors. What is its output?

```c
#include <stdio.h>

int ack(int, int);

void main(void)
{
    int m = 1;
    int n = 1;

    printf("%i \n", ack(m, n));
}

int ack(int m, int n)
{
    if (m == 0)
        return n+1; /* VOOM!! */
    else if (m > 0 && n == 0)
        return ack(m-1,1);
    else
        return ack(m - 1, ack(m - 1, ack(m, n - 1)));
}

(a) 2  
(b) 3  
(c) 4  
(d) 5  
```
25. The recursive program below compiles and runs without errors. What is its output?

```c
#include <stdio.h>

void sierp(int, int, int *);

void main(void)
{
    int n = 2;
    int max = 3;
    int count = 0;

    sierp(n,max,&count);

    printf("%i \n",count);
}

void sierp(int n, int max, int * count)
{
    ++(*count);

    if (n >= max)
        return;

    sierp(n + 1, max, count);
    sierp(n + 2, max, count);
    sierp(n + 3, max, count);
}

(a) 3
(b) 4
(c) 2
(d) none of the above
```
Extra Credit

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

26. The recursive program below compiles and runs without errors. What is its output?

```c
#include <stdio.h>

int p(char * str, int len)
{
    if(len <= 1)
        return 1;
    else if(str[0] != str[len-1])
        return 0;
    else
        return p(str+1, len-2);
}

void main(void)
{
    printf("%i %i %i \n", p("ada", 3), p("madam", 5), p("madame", 6));
}
```

(a) 0 0 0
(b) 1 1 1
(c) 1 1 0
(d) 0 0 1

27. For the following structure,

```c
typedef struct
{
    char title[50];
    char author[50];
    int pub_date;
} Book;
```

and the declaration,

```c
    {"Quantum Physics","Gasiorowicz",1974}};
```

Which of the following expressions, when substituted for the blank in the C statement below, will NOT result in printing the title of the first element in the bookarr array? (We assume that the code above is also included in a complete C program that compiles and runs without errors.)

```c
printf("%s", ______________);
```

(a) bookarr.title[0]
(b) bookarr[0].title
(c) (*bookarr).title
(d) bookarr->title