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></td>
<td></td>
<td>101</td>
</tr>
<tr>
<td>10:00-11:50</td>
<td>102</td>
<td>102</td>
<td></td>
<td></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.
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. Assume that you have a file (containing a C program) named `lab7.c` in your current working directory. Which of the following series of commands, typed at the Unix prompt, will successfully compile and run the C program?

(a) `gcc lab7.c -lm lab7
   ./lab7`

(b) `gcc lab7.c
   ./lab7.c`

(c) `gcc lab7.c
   ./a.out`

(d) `gcc lab7.c -o lab7
   ./lab7.c`

For the following TWO questions assume that you have the following directory structure.

2. Assume that your Unix home directory is named `cs101stu` and that your current working directory is named `cs101ta`. You wish to change your current working directory to the directory named `labs`. Which of the following Unix commands will do this?

(a) `cd ~/labs`

(b) `cd home/cs101stu/labs`

(c) `cd /labs`

(d) `cd ..home/cs101stu/labs`

3. Assume that your Unix home directory is named `cs101stu` and that your current working directory is named `usr`. You wish to delete the `mps` directory and all its contents. Which of the following Unix commands will do so?

(a) `del -r ~/mps`

(b) `rm ~/cs101stu/mps`

(c) `rm ~/mps`

(d) `rm -r /home/cs101stu/mps`
4. Will the following C program compile, and run, successfully? If so, what is its output?

```c
#include <stdio.h>

void main(void)
{
    int x = 5;
    if (x < 5);
        printf("A \n");
    else if(x == 6)
        printf("B \n");
        printf("C \n");
    else
        printf("D")
}
```

(a) A
(b) D
(c) A
    C
(d) No, the program does not compile successfully

5. Which of the following is an illegal C declaration-initialization?

(a) `char y;
   char * ptr1 = &y;`

(b) `int arr[4] = {1, 2, 3};`

(c) `char * word = "cat";`

(d) `double * addresses[][][10];`
6. The following C program compiles and runs without errors. What is the output of the program?

```c
#include <stdio.h>

void main(void)
{
    int x = 2, y = 0, option = 2, password;

    password = !(x == y);

    if(password)
        x = y;
    else if(x == y)
    {
        switch(option)
        {
            case 1: x = 1;
                break;
            case 2: x = 2;
            case 3: x = 3;
                break;
            default: x = 4;
        }
    }
    else
        printf("%i",x);

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

(a) 3 0 2

(b) 0 0 2

(c) 2 0 2

(d) 2 2 0 2

7. Which of the following lines of C code will define a new data type named string? The correct answer choice would enable you to correctly declare and initialize the following variable named x.

```c
string x = "kitty cat";
```

(a) typedef string char[32];

(b) typedef string[32] char;

(c) typedef char string[32];

(d) typedef char[32] string;
8. Which of the following programs prints the transpose of the following matrix?

\[
\begin{bmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9 \\
\end{bmatrix}
\]

\[
\text{the transpose is}
\begin{bmatrix}
1 & 4 & 7 \\
2 & 5 & 8 \\
3 & 6 & 9 \\
\end{bmatrix}
\]

(a) #include <stdio.h>

void main(void)
{
    int row, col;
    int arr1[3][3] = {{1,2,3},
                      {4,5,6},
                      {7,8,9}};

    for(row = 0; row < 3; ++row)
        for(col = 0; col < 3; ++col)
            arr1[col][row] = arr1[row][col];

    col = 0;
    for(row = 0; row < 3; ++row)
        printf("%i %i %i \n",arr1[row][col],
                      arr1[row][col+1],
                      arr1[row][col+2]);
}

(b) #include <stdio.h>

void main(void)
{
    int row, col;
    int container[3][3];
    int arr1[3][3] = {{1,2,3},
                      {4,5,6},
                      {7,8,9}};

    for(row = 0; row < 3; ++row)
        for(col = 0; col < 3; ++col)
            container[col][row] = arr1[row][col];

    col = 0;
    for(row = 0; row < 3; ++row)
        printf("%i %i %i \n",container[row][col],
                        container[row][col+1],
                        container[row][col+2]);
}

(This question is continued on the next page.)
(c) #include <stdio.h>

void main(void)
{
    int row, col;
    int container[3][3];
    int arr1[3][3] = {
        {1,2,3},
        {4,5,6},
        {7,8,9}};

    for(row = 0; row < 3; ++row)
        for(col = 0; col < 3; ++col)
            arr1[row][col] = arr1[col][row];

    col = 0;
    for(row = 0; row < 3; ++row)
        printf("%i %i %i \n",arr1[row][col],
            arr1[row][col+1],
            arr1[row][col+2]);
}

(d) #include <stdio.h>

void main(void)
{
    int row, col, temp;
    int container[3][3];
    int arr1[3][3] = {
        {1,2,3},
        {4,5,6},
        {7,8,9}};

    for(row = 0; row < 3; ++row)
        for(col = 0; col < 3; ++col)
            {
                temp = arr1[row][col];
                arr1[col][row] = arr1[row][col];
                arr1[row][col] = temp;
            }

    col = 0;
    for(row = 0; row < 3; ++row)
        printf("%i %i %i \n",arr1[row][col],
            arr1[row][col+1],
            arr1[row][col+2]);
}
9. True or False, the following program, which will compile and run without errors, swaps the values stored in the variables x and y in the function main (i.e. the program will print 2 1).

```c
#include <stdio.h>

void swap(int *, int *);

void main(void)
{
    int x = 1, y = 2;
    swap(&x,&y);
    printf("%i %i 
",x,y);
}

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

(a) True
(b) False
```

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

```c
#include <stdio.h>

int i = -10;

int function2(int x)
{
    int i = -12;
    printf(" %i ", i);
    return x * i;
}

void main(void)
{
    printf(" %i ", function2(i));
    printf(" %i\n", i);
}

(a) -12 120 -10
(b) -12 120 -12
(c) -10 100 -12
(d) -12 144 -10
```
11. The following C program compiles and runs without errors. What is the output of the program?

```c
#include <stdio.h>

void main(void)
{
    int x[4] = {10, 20, 30, 40};
    int *ptr;
    ptr = x;
    ++$ptr;

    if(x[0] == *ptr)
    {
        printf("%i \n", *ptr);
        ++ptr;
    }

    printf("%i \n", *ptr);
}
```

(a) 10
    20

(b) 20
    30

(c) 11
    20

(d) 11
12. The code fragments below are each copied into separate complete C programs that compile and run without errors. Which of the following code fragments will **NOT** yield the following output?

1 2 3 4

(a) int x = 1;
   for(x = 1; x < 5; x++)
      printf("%i ",x);

(b) int x = 1;
    do {
       printf("%i ",x);
    }while(++x<5);

(c) int x = 1;
    for(x = 1; x < 5; ++x)
       printf("%i ",x);

(d) int x = 1;
    while( ++x<5)
       printf("%i ",x);

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

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

void main(void)
{
    char word1[20] = "sandwich";
    char word2[20] = {'s','a','n','d','w','i','c','h','\0'};

    printf("%i \n",strcmp(word1,word2));
    printf("%s \n",word1);
    printf("%s \n",word2);
}
```

(a) 1
    sandwich
    sandwich\0

(b) -1
    sandwich
    sandwich

(c) 0
    sandwich
    sandwich

(d) -1
    sandwich
    sandwich\0
14. How many distinct variables are declared in the program below?

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

#define VAL 2.2

void func1(int *a, int y);
void func2(int a, int b);
void func3(double a, double b, double c);

int x,y,z;

void main(void)
{
    double a = 1.1, b = VAL, c = 3.3;
    x = 1;
    y = 3;
    z = 5;

    func1(&x,y);
    func2(x,y);
    func3(a,b,c);
}

void func1(int *a, int y)
{
    y = 3;
    x = 4;
    printf("%i", y);
}

void func2(int a, int b)
{
    b = x + y + z;
    printf("%i %i", a, b);
}

void func3(double a, double b, double c)
{
    printf("%lf %lf %lf \n", a, b, c);
}

(a) 7
(b) 19
(c) 24
(d) 13
```
15. True or False. The auxiliary function named helper shown below is used with qsort to sort the elements of array named arr in descending order.

```c
#include <stdio.h>

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

void main(void)
{
    int arr[8] = {4,7,2,3,9,1,5,10};
    qsort(arr,8,sizeof(int),helper);
}
```

(a) True
(b) False

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

```c
#include <stdio.h>

void func(int x[])
{
    x[0] = x[0]+1;
}

void main(void)
{
    int arr[] = {3,4,5,6};

    func(&arr[1]);
    printf("%i %i \n",arr[1],arr[2]);
}
```

(a) 3 4
(b) 4 5
(c) 4 4
(d) 5 5
17. Assume that the data file named input.dat has the following data:

\[-1\]

that is, a single value, minus one. Further assume that the C program shown below is in the file named q17.c and compiles without errors and is run by typing the following at the Unix prompt:

```
./a.out < input.dat
```

What is its output?

```
#include <stdio.h>

void main(void)
{
    int value = 999, response = 42;
    response = scanf("%i", &value);
    printf("%d %d \n", value, response);
}
```

(a) \(-1 -1\)
(b) \(-1 999\)
(c) \(-1 1\)
(d) \(1 999\)

18. Which answer choice below that when replacing the blank below would NOT lead to an infinite loop?

```
#include <stdio.h>

void main(void)
{
    int a;
    a=0;

    ______(answer)________________________
    {
        a=a+1;
    }
    printf(" a = %i \n", a);
}
```

(a) for( ; ; --a)
(b) for( ; a<10 ; a++ )
(c) for( a=0 ; ; a--)
(d) for( a=0; a<10; a-- )
Given the information shown below, answer the next two questions.

The variables S, ptr1 and ptr2 and array arr have been initialized in memory as shown in the code fragment below. The memory addresses that they occupy are given in the table below.

<table>
<thead>
<tr>
<th>variable</th>
<th>memory address</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>3000</td>
</tr>
<tr>
<td>arr[0]</td>
<td>1000</td>
</tr>
<tr>
<td>arr[1]</td>
<td>1004</td>
</tr>
<tr>
<td>arr[2]</td>
<td>1008</td>
</tr>
<tr>
<td>ptr1</td>
<td>4000</td>
</tr>
<tr>
<td>ptr2</td>
<td>4004</td>
</tr>
</tbody>
</table>

```c
int S = 7;
int arr[] = {1,2,3};
int * ptr1 = &S;
int * ptr2 = arr;
*ptr1 = 4;
ptr1 = ptr2+1;
*(ptr1+1) = *ptr2;
```

19. What are the three values stored in arr[0], arr[1] and arr[2] after the above code fragment is executed?

   (a) \{1,2,1\}

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

   (c) \{1,3,3\}

   (d) \{2,2,3\}

20. What is the final value of ptr1?

   (a) 1004

   (b) 1000

   (c) 4000

   (d) 1008
21. The following program generates random integers between 1 and 10. It does so by asking the user how many random numbers they want at program run-time. The program must then allocate enough memory for whatever number of integers the user desires and then generates them. Select the expression that, when inserted in place of the blank in the program below, will correctly allocate enough memory for the random integers and allow the program to compile and run without errors.

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

int * numberLine(int num)
{
    int i;

    _______________________________________________________________
    if(points == NULL)
    {
        printf("Not enough memory available");
        return points;
    }
    srand(time(NULL));
    for (i = 0; i < num; ++i)
        points[i] = rand()%10 + 1;
    return points;
}

void main(void)
{
    int num, i;
    int * points;

    printf("How many random integers? ");
    scanf("%i", &num);
    points = numberLine(num);
    if(points == NULL)
    {
        printf("Exiting... \n");
        return;
    }
    for(i = 0; i < num; ++i)
        printf("%i ", *(points + i));
    free(points); /* always free the memory to prevent a memory leak */
}

(a) int points[] = calloc(num,sizeof(int));
(b) int * points = malloc(num);
(c) int * points = calloc(num,sizeof(int));
(d) int points = calloc(sizeof(int),num);
```
22. Which of the following statements is TRUE about the following code fragment?

```c
FILE * ptr;

ptr = fopen("input.dat","r");

if(ptr == NULL)
{
    printf("File does not exist!!");
    return;
}
```

(a) The code fragment attempts to open the file **input.dat** and if the file does not exist it creates it.

(b) There is a syntax error in the call to **fopen** and the file **input.dat** is not successfully opened in read mode.

(c) **FILE** is not a valid C data type, which would result in a compile error.

(d) Trapping an error (the purpose of the **if** statement above) is necessary because if the file **input.dat** does not exist, we need to terminate the function.
23. You are given a text file named `input.dat` containing the following data:

Mars, July. 1st 2030, 14, 30, 12, 1.524, 0.093, 1.849, 49.523, 286.605, 95.642

You are also given the following incomplete program that is supposed to read the text file and then print to the computer screen the following output.

Mars July. 1st 2030 14:30:12
a: 1.524
e: 0.093
i: 1.849
RAN: 49.523
omega: 286.605
M: 95.642

```c
#include <stdio.h>

typedef struct{
    char name[20];
    char date[20];
    int hour;
    int minute;
    int second;
    double a;
    double e;
    double i;
    double RAN;
    double omega;
    double M;
} TLE;

void scanner(char * file, TLE * p)
{
    FILE * fileptr = fopen(file,"r");
    fscanf(fileptr, ________________________,
            p->name, p->date, &p->hour, &p->minute, &p->second,
            &p->a, &p->e, &p->i, &p->RAN, &p->omega, &p->M);
}

void main(void)
{
    TLE redPlanet;
    char filename[] = "input.dat";
    scanner(filename, &redPlanet);
    printf("%s %s %i:%i:%i 
",
            redPlanet.name, redPlanet.date,
            redPlanet.hour, redPlanet.minute, redPlanet.second);
    printf("a: %.3lf 
e: %.3lf 
i: %.3lf 
RAN: %.3lf 
omega: %.3lf 
M: %.3lf",
            redPlanet.a, redPlanet.e, redPlanet.i,
            redPlanet.RAN, redPlanet.omega, redPlanet.M);
}
```

Select the code fragment below, that when replacing the blank in the `fscanf` above, will allow the program to read the data properly and compile and run without errors.

(a) "[^;], %[^;], %i, %i, %i, %lf, %lf, %lf, %lf, %lf, %lf"
(b) "[^;], %[^;], %i, %i, %i, %lf, %lf, %lf, %lf, %lf, %lf"
(c) "%c, %[^;], %i, %i, %i, %lf, %lf, %lf, %lf, %lf, %lf"
(d) "%c, %c, %i, %i, %i, %lf, %lf, %lf, %lf, %lf, %lf"
The next two questions use definitions below.

```c
#define MONTHS 12

typedef struct {
    char   name[40]; /* track name or song name */
    int    length;  /* in seconds */
} Track;

typedef struct {
    int    cdno;    /* cd number */
    char   title[30];
    char   artist[20];
    int    year;
    int    num_tracks; /* number of tracks */
    int    quantity;
    float  price;
    float  sales[MONTHS]; /* cd sales ($US) over the 12 months in 2013 */
} Album;

Track tracks[20]; /* info for each track */
```

Using the definitions above, assume that the following array, and a pointer to that array have been declared.

```c
Album albums[100];
Album * ptr = albums;
```

24. Which of the following options will print out all of the entries of the `sales` array of the 20th album in the `albums` array (assuming the `sales` array field has been initialized)?

(a) `printf("%f ", albums[19].sales);`
(b) `int i;
    for(i = 0; i < sizeof(albums[20].sales); ++i)
        printf("%f ", albums[20].sales[i]);`
(c) `int i;
    for(i = 0; i < MONTHS; ++i)
        printf("%f ", albums[19].sales[i]);`
(d) `printf("%s 
", albums[19].sales);`

25. You want to assign "One Heart" to the `name` field of the 7th track of the 6th album in the `albums` array and then prints the string. Which of the following code fragments will NOT do this?

(a) `album[5].tracks[6].name = "One Heart";
    printf("%s 
", albums[5].tracks[6].name);`
(b) `strcpy((ptr+5)->tracks[6].name,"One Heart");
    printf("%s 
", albums[5].tracks[6].name);`
(c) `int i;
    char word[40] = "One Heart";
    for(i = 0; i <= strlen(word); ++i)
        albums[5].tracks[6].name[i] = word[i];
    printf("%s 
", albums[5].tracks[6].name);`
(d) `strcpy((*(ptr+5)).tracks[6].name,"One Heart");
    printf("%s 
", albums[5].tracks[6].name);`
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.

The assigned values for $S$, $ptr1$ and $ptr2$ are given below. Given that the address in memory of $S$ is 3000, the address of $ptr1$ is 3004 and the address of $ptr2$ is 3008 answer the following question.

```c
int S = 7;
int * ptr1 = &S;
int * *ptr2 = &ptr1;

*ptr1 = 4;
*ptr2 = ptr1;
*(ptr2) = 8;
```

26. What is the final values of $S$ and $*ptr2$ respectively, when the above code placed in a complete C program that compiles and runs without errors?

(a) 4 ; 3000  
(b) 4 ; 3008  
(c) 8 ; 3000  
(d) 8 ; 3004

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

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

void main(void)
{
    char str[] = "elephant";

    strcpy(str, "room");

    if( strstr(str, "ant") != NULL)  
        {  
            printf("Yes ");
            return;
        }  
    else if(strstr(str, "room") == NULL)  
        {  
            printf("No ");
            return;
        }  
    printf("Maybe ");
}
```

(a) Yes  
(b) No  
(c) Maybe  
(d) Yes Maybe