

NAME _____ S.S.# _____

#1. (Functions) Write a function that receives 3 integers, and returns the value of the largest one through the function name.

#2. (Functions) What is the output of this segment of code.

```
int z = 0;
```

Ans.

```
void bill (int c)
{
    int x = 0;
    static int y = 0;
    x = x + c;
    y = y + 2 * c;
    z = z + 3 * c;
    printf (" x = %d, y = %d, z = %d \n",x,y,z);
}
```

```
main()
{
    int x = 5;

    bill (x);
    x = x + 2;
    bill (x);
    bill (z);
}
```

#3. (Recursion) Write a recursive function to compute $f(x) = f(x - 1) + f(x - 2)$ with $f(0) = 0$ and $f(1) = 1$.

#4. (Arrays) Write a loop to double the value of each component of an array of integers called A. Stop at the component containing 0. Ex. before: $A = \{2,5,7,0,3,5\}$ after $A = \{4,10,14,0,3,5\}$. Declare the array A and any variable needed.

#5. (Structures and functions) Declare a structure called point that stores 2 integers, x and y. Declare another structure called segment that stores 2 points, p1 and p2. Finally write a function that given a segment and a point determines if the point is on the segment. The function returns true if the point is on the segment and false otherwise. A point is on a segment if the distance from one end of the segment, p1, to the other end, p2, is the same as the distance from p1 to the point plus the distance from the point to p2.

#6. (Pointers) What is the output of the following code segment. Assume the address of x is 4000 and y is 4002.

```
void bill(int *x,*y)
{
*x = *x + 10;
*y = *y + 20;
y = 8000;
}
```

Ans.

```
void main()
{
int x,y,*p;

x = 0;
y = 5;
bill(&x,&y);
printf(“%d %d\n”,x,y);

bill(&y,&x);
printf(“%d %d \n”,x,y);

p = 4000;
bill(p,p);
printf(“%d %d\n”,x,y);
}
```

#7. (Scope). On the right side show what is the output of this program.

```
#include "stdio.h"

int b[5] = {1,2,1,2,5},
    c[5] = {2,1,2,1,0};

void
bill(int x, int *p
    {
    int I;

    x--;
    for (I = 0; I < x; I++)
        p[I] = p[I] + b[I] + c[I];
    }

void
main()
    {
    int a[5] = {1,2,3,4,5},
        c[5] = {5,4,3,2,1};

    int I;

    bill(a[4],a);
    for (I = 0; I < 5; I++)
        printf(" %d ",a[I]);
    printf("\n");

    bill(c[0],c);
    for (I = 0; I < 5; I++)
        printf(" %d ",c[I]);
    printf("\n");

    bill(b[4],b);
    for (I = 0; I < 5; I++)
        printf(" %d ",b[I]);
    printf("\n");
    }
```

#8. (Scope) What is the output if you enter 1,2 and 3 for k:

```
void main()
{
    int    *p,x;           (k == 1)  _____
    char   c;              (k == 2)  _____

    p = &x;
    x = 3;                 (k == 3)  _____

    y = 4;

    scanf("%d", &k);

    if (k == 1)
        {
            int x;

            x = 5;
            y = 6;
        }
    else if (k == 2)
        {
            int x;

            *p = 7;
            x = 8;
            y = 9;
        }
    else
        {
            x = 10;
            y = 11;
        }

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

- #9. (Functions and Scope) What is the output of the following code segment. Assume the address of x is 4000 and y is 4002.

```
void bill(int x,*y)           Ans.
{
    x = 56;
    *y = 72;
    y = 8000;
}
```

```
void main()
{
    int x,y,*p,a[3];

    x = 37;
    y = 42;
    bill(x,&y);
    printf(“%d %d\n”,x,y);

    p = 4000;
    bill(x,p)
    printf(“%d %d\n”,x,y);

    a[0] = 1;a[1] = 2; a[2] = 3;
    bill(a[0],a);
    printf(“%d %d %d \n”,a[0],a[1],a[2]);
}
```

- #10. (Just a good exercise) Write a for loop to print the first 10 numbers in the sequence where the next number is the previous number plus the index that is being incremented by 1 each time it goes through the loop. ex 0 1 3 6 10 15 21 ... note: $0+(0) = 0$, $0+(1) = 1$, $1+(2) = 3$, $3+(3) = 6$, $6+(4) = 10$, ...

- #11. (Recursion) Write a recursive function to compute $f(x) = f(x - 1) + x$ with $f(0)= 0$. This will generate the sequence in the problem above.

- #12. (Files) Write a function to read the data from a file called data.dat into a 2 dimensional array of integers. The first line in the file contains 2 numbers, the row and column dimensions respectively. The remainder of the lines in the file contain the data for the array organized by its dimensions (each row is on a new line).

Example data.dat file: **Do not assume the file is as below.**

```
2 3
1 5 2
4 6 1
```

You must dynamically allocate memory for this array since you do not know ahead of time how large this array is to be.

Hint: open the file, read the first line (row, column), dynamically allocate sufficient memory using malloc or new, then finally use a double “for” loop to initialize the array from the file.

- #13. (Short answers). Answer each of the following.
- What does extern do.
 - What does static do. Note it does 2 different things depending on how you use it.
 - What must one do to be able to modify a variable in a function and have the change be reflected in the main program that calls the function. Give example.
 - What is the difference between using fprintf and fwrite. Give a typical application for using each.
 - What is the advantage of using pointers and dynamic memory allocation when implementing a stack or a queue? Why not just use an array?
- #14. (Declarations) Declare the following variables. Do not initialize them.
- Declare a structure called record that holds a person's name, age, social security number phone number and salary.
 - Declare a structure called node to be used in a linked list that has pointers to the previous and next nodes and the data part is a pointer to a void.
 - Declare an array with 5 elements where each element is a pointer to a double called apd.
 - Declare an array of 5 elements where each element is an array of 3 elements of doubles called aad.

