

NAME Solutions SS# \_\_\_\_\_

To receive maximum partial credit, clearly show all of your work.

1. (Functions) Write a function to perform division. The function is to accept 3 doubles, the numerator and the denominator and a pointer to the remainder. The function is to divide the 2 doubles and produce a quotient and a remainder. The quotient is to be returned through the function name and the remainder through the pointer in the third parameter. Assume you have the functions floor ( ) which returns the integer part of a double and the function frac ( ) which returns the fractional part of a double.

Algorithm:

Result = numerator / denominator;

Quotient = floor(result)

Remainder = frac(result) \* denominator

```
double divide(double num, double denom,  
              double *rem)
```

```
{
```

```
double Quo, Result;
```

```
Result = num / denom;
```

```
Quo = floor(Result);
```

```
*Rem = fract(Result) * denom;
```

```
return Quo;
```

```
}
```

2. (File) Write a program to open two files called "input1.dat" and "input2.dat" both for input. Read the first 5 integers from each file and compute the sum of the 10 integers. Print the sum then close the files. Print an error message if the file cannot be opened.

```
FILE *fp1, fp2;
```

```
main()
```

```
{  
    int sum = 0; X;
```

```
    fp1 = fopen("input1.dat", "r");
```

```
    fp2 = fopen("input2.dat", "r");
```

```
    if (fp1 == null || fp2 == null)  
        printf("error opening files\n");
```

```
    else
```

```
    {  
        for (X = 0; X < 5; X++)
```

```
        {  
            fscanf(fp1, "%d", &X);
```

```
            sum = sum + X;
```

```
        }  
        for (X = 0; X < 5; X++)
```

```
        {  
            fscanf(fp2, "%d", &X);
```

```
            sum = sum + X;
```

```
        }  
    }
```

```
printf("The sum is %d \n", sum);
```

```
fclose(fp1);
```

```
fclose(fp2);
```

```
} // end if
```

```
} // end main
```

3. (Classes) Create a class called Point that has as private elements the x and y positions on the Cartesian graph. The class is to have as public the color code of the point represented by an integer, a function that returns the distance the point is from the origin, use the formula  $dist = \sqrt{x^2 + y^2}$ , and a constructor that initializes the point's x and y position and sets the color to 0.

```
class Point
{
private:
    double X;
    double Y;
public:
    int color;
    double dist()
    {
        return sqrt(X*X + Y*Y);
    }
    Point(double newX, double newY)
    {
        X = newX;
        Y = newY;
        color = 0;
    }
}
```





6. (Short answers). Answer each of the following.  
Assume a = 3, b = 7, c = 17.

```
for (x = -1; x < 2; x++)
    for (y = 0; y < 2; y++)
        printf("%d %d\n", x, y);
```

-1, 0  
-1, 1  
0, 0  
0, 1  
1, 0  
1, 1

```
for (x = 0; x < 8; x = x + 3)
{
    y = 1;
    while (y < x)
    {
        printf(" %d %d \n ", x, y);
        y = y * 2;
    }
}
```

3, 1  
3, 2  
6, 1  
6, 2  
6, 4

```
for (x = 0; x < 6; x++)
    for (y = 0; y < 10; y++)
        if (x + y == 10)
            printf("%d %d\n", x, y);
```

1, 9  
2, 8  
3, 7  
4, 6  
5, 5

```
if ((a == 3 && b < 10) || (a == 4 && b > 10))
    printf("Red \n");
else
    printf("Blue \n");
```

Red

```
if (a == 3 || b < 4)
    if (b == 5 && c < 10)
        printf("Yellow \n");
    else
        printf("Green \n");
else
    if (b == 15 && c > 10)
        printf("Orange \n");
    else
        printf("Black \n");
```

Green

X Y  
0 1  
3 2  
6 4  
9 1  
2  
4  
8

7. (Declarations) Declare the following variables. Do not initialize them.

- a. Declare a structure called address with 3 fields of type string called street, city, and state.

```
struct address
{
    char street[80];
    char city[80];
    char state[80];
}
```

- b. Declare an array of 3 by 5 where each element is a pointer to struct address called aapsa.

```
address *aapsa[3][5];
```

- c. Declare 2 objects of type Point as defined in problem 4 called p1 and p2. The object p1 is to be initialized to 5 and 3 and p2 to 10 and 7.

```
Point p1(5,3), p2(10,7);
```

- d. Declare a 3 dimensional array, 3 by 5 by 10 of integers called aaai.

```
int aaai[3][5][10];
```

- e. Declare an array with 10 elements where each element is an 80-character string called as.

```
char as[80][10]
```