CS 1713
Introduction to Computer Programming II
Midterm

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question 1</td>
<td></td>
</tr>
<tr>
<td>Question 2</td>
<td></td>
</tr>
<tr>
<td>Question 3</td>
<td></td>
</tr>
<tr>
<td>Question 4</td>
<td></td>
</tr>
<tr>
<td>Question 5</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
</tr>
</tbody>
</table>

NAME:____________________

Instructions
1. Do all of the 5 problems
3. You have 70 minutes for the exam
4. Show all your work
5. Do not separate midterm papers

Easy       Difficulty Level         Difficult
☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐
1  2  3  4  5  6  7  8  9  10
1. (20 pts) Complete the following program to find the average and standard deviation of the numbers in an array. For example, the array \{2, 4, 6, 3, 9, 10\} has standard deviation of average of 5.666667 and standard deviation of 2.981424. Formula for standard deviation is as follows.

\[
\text{stddev} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2}
\]

In the formula \(n\) denotes the number of items in the set, \(x_i\) denotes \(i^{th}\) number and \(\bar{x}\) denotes the average of the numbers in the set.

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

int main()
{
    double stddev; //standard deviation
    int i;
    double num[10]; // array to store the numbers
    double average, double sum=0;

    printf("Enter 10 doubles\n");
    for (i=0; i<10; i++)
    {
        scanf("%lf", &num[i]);
        for (i=0; i<10; i++)
        {
            sum = sum + num[i];
        }
        average = sum / 10;
        sum = 0;
        for (i=0; i<10; i++)
        {
            sum = sum + (num[i] - average) * (num[i] - average);
        }
        stddev = sqrt(sum / 10);
    }

    printf("Average is \%lf\n",average);
    printf("Standard Deviation is \%lf\n",stddev);
    return(0);
}
```
2. (20 pts) Trace the execution of the following program? What will be the final values of array a printed for the cases given below.

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

int main()
{
    int a[7]=
    {1, 1, 0, 0, 0, 0, 0};
    int i=2;

    while (i<7)
    {
        if (i<5)
            a[i] = a[i-1]+a[i-2];
        else
            a[i] = a[i-3];
        i = i + 1;
    }

    for (i=0; i<7; i++)
        printf("a[%d] = %d\n",i,a[i]);
}
```

(a) What will be the final array when initial array is `a[7] = {1, 1, 0, 0, 0, 0, 0}`?

```
```

(a) `a[7] = {1, 1, 0, 0, 0, 0, 0}`

(b) What will be the final array when initial array is `a[7] = {4, 2, 3, 6, 4, 2, 5}`?

```
a[0] = 1
a[1] = 1
a[2] = 2
a[3] = 3
a[4] = 5
a[5] = 2
a[6] = 3
```

(b) `a[7] = {4, 2, 3, 6, 4, 2, 5}`
3. (20 pts) Write a function `Powerof3` to test if a parameter \( n \) is a power of 3 \( (n = 3^k \) for some integer \( k \)). If \( n \) is a power of 3, then the function returns 1. Otherwise it returns 0. Function prototype and sample output of the function and description of the output is given below:

for \( n=5 \) function returns 0 since \( 3^1 = 3 < 5 \) < \( 3^2 = 9 \)
for \( n=9 \) function returns 1 since \( 9 = 3^2 \)
for \( n=30 \) function returns 0 since \( 3^3 = 27 < 30 < 3^4 = 81 \)
for \( n=27 \) function returns 1 since \( 27 = 3^3 \)

```c
int Powerof3(int n)
{
    int (X), power;
    power = 1;
    while (power < n)
    {
        i=i+1;
        power = power * 3;
    }
    if (power == n)
        return 1;
    else
        return 0;
}
```

<table>
<thead>
<tr>
<th>( n )</th>
<th>( 3 )</th>
<th>( 9 )</th>
<th>( 27 )</th>
<th>( 81 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 3 )</td>
<td>( 9 )</td>
<td>( 27 )</td>
<td>( 81 )</td>
<td></td>
</tr>
<tr>
<td>( 3 )</td>
<td>( i )</td>
<td>( 3 )</td>
<td>( i )</td>
<td>( )</td>
</tr>
</tbody>
</table>

```c
while (power < n)
{
    i=i+1;
    power = pow(3,i);
}
```
4. (20 pts) What is the output of the following program? Show all your work for partial credit.

```c
#include <stdio.h>

int function1(int x) {
    return(2*x+1);
}

int main() {
    int j=3;
    while (j < 50) {
        if (j<10)
            j = function1(j-1);
        else
            j = function1(j)-1;
        printf("%d\n",j);
    }
    return(0);
}
```

The output is:

5
9
17
34
68

The calculation is as follows:

- \( j = \text{function1}(2) = 5 \)
- \( j = \text{function1}(4) = 9 \)
- \( j = \text{function1}(8) = 17 \)
- \( j = \text{function1}(17) - 1 = 34 \)
- \( j = \text{function1}(34) - 1 = 68 \)
5. (20 pts) Write a complete program to compute the following expression. Read the value of \( n \) from the user and write a loop to evaluate the product. \( \prod \) denotes multiplication of all the terms.

\[
\frac{1}{2 \times 2 - 1} \times \frac{2}{2 \times 2 - 1} \times \cdots \times \frac{i}{2 \times i - 1} \times \cdots \times \frac{n}{2 \times n - 1} = \prod_{i=1}^{n} \frac{i}{2i - 1}
\]

```c
#include <stdio.h>

int main ()
{
    int n, i;
    float product = 1;
    printf("Enter n \n");
    scanf("%d", &n);
    for (i = 1; i <= n; i++)
        product = product * i / (2 * i - 1);
    printf("Product = \%f \n", product);
    return (0);
}
```