# 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><strong>Total</strong></td>
<td></td>
</tr>
</tbody>
</table>

**NAME:** __________________________

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

<table>
<thead>
<tr>
<th>Easy</th>
<th>Difficulty Level</th>
<th>Difficult</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td>□□□□□□□□□□□□□□□</td>
<td>□□□□□□□□□□</td>
</tr>
<tr>
<td>1</td>
<td>2    3</td>
<td>4  5  6  7  8  9  10</td>
</tr>
</tbody>
</table>
1. (20 pts) Complete the following program to find the closest number to average in the array.
   For example, the array \{2, 4, 6, 3, 9, 10\} has average of 5.666667 and closest number to average in the array is 6. Note that closest number can be larger than or smaller than the average.

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

int main()
{
    int i;
    double num[6];
    double average;
    double sum=0;
    double closest;

    printf("Enter 6 doubles\n");
    for (i=0; i<6; i++)
    {
        scanf("%lf",&num[i]);

        for (i=0; i<6; i++)
            sum = sum + num[i];
    }

    average = sum / 6;

    closest = num[0];
    for (i=0; i<6; i++)
        if (fabs(num[i]-average) < fabs(closest-average))
            closest = num[i];

    printf("Closest is %lf\n",closest);
    return(0);
}
```
2. (20 pts) Trace the execution of the following program? What will be the final values of array a printed?

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

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

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

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

```
a[0] = 2
a[1] = 4
a[2] = 7
a[3] = 5
a[4] = 8
a[5] = 23
a[6] = 46
```

```
a[1] = a[0] + 1 + 1
a[2] = a[2-1] + 2 + 1
           = a[1] + 3
           = a[1] + 1
a[4] = a[4-2] + 1
           = a[2] + 1
           = a[4] * 6 - 2
```
3. (20 pts) Write a function `Powerof` to test if a parameter \( n \) is a power of another parameter \( k \) (\( n = k^m \) for some integer \( m \)). If \( n \) is a power of \( k \), 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:

- `Powerof(3,5)` returns 0 since \( 3^1 = 3 < 5 < 3^2 = 9 \)
- `Powerof(3,9)` returns 1 since \( 9 = 3^2 \)
- `Powerof(2,30)` returns 0 since \( 2^4 = 16 < 30 < 2^5 = 32 \)
- `Powerof(2,16)` returns 1 since \( 16 = 2^4 \)

```c
int Powerof(int k, int n) {
    int i = 1, power;
    power = k;
    while (power < n) {
        i = i + 1;
        power = power * k;
    }
    if (power == n)
        return (1);
    else
        return (0);
}
```
4. (20 pts) What is the output of the following program? Show all your work for partial credit.

```c
#include <stdio.h>

int mystery(int a, int b)
{
    return(a*b+1);
}

int mystery2(int a, int b)
{
    return(a*(b+1));
}

int main()
{
    int j,k;
    for (j=0; j<3; j++)
    {
        if (j<3)
            k = mystery(j+1,j);
        else if (j<6)
            k = mystery2(j+1,j);
        else
            k = mystery(mystery2(j,j),1);
        printf("j=%d k=%d\n",j,k);
    }
    return(0);
}
```

```

<table>
<thead>
<tr>
<th>j</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>25</td>
</tr>
<tr>
<td>5</td>
<td>36</td>
</tr>
<tr>
<td>6</td>
<td>43</td>
</tr>
<tr>
<td>7</td>
<td>57</td>
</tr>
</tbody>
</table>
```

```

\[ k = \text{mystery}(1, 0) = 1 \]
\[ k = \text{mystery}(2, 1) = 3 \]
\[ k = \text{mystery}(3, 2) = 7 \]
\[ k = \text{mystery2}(4, 3) = 16 \]
\[ k = \text{mystery2}(5, 4) = 25 \]
\[ 5 \times (4+1) = 25 \]
\[ k = \text{mystery2}(6, 5) \]
\[ 6 \times (5+1) = 36 \]
\[ k = \text{mystery}(\text{mystery2}(6, 6), 1) \]
\[ 42 \times 1 + 1 = 43 \]
\[ k = \text{mystery}(\text{mystery2}(7, 7), 1) \]
\[ 56 \times 1 + 1 = 57 \]
```
5. (20 pts) Write a complete program to compute the smallest value of $k$ for which the following sum is larger than $n$. Read the value of $n$ from the user and write a loop to compute the sum. Note that $n$ can be a floating point number.

$$\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{k} > n$$

For example for $n = 2.0$, Your program should find $k$ to be 4 since

$$\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} = 2.08 > 2$$

For $n = 1.6$, your program should find $k$ to be 3 since $\frac{1}{1} + \frac{1}{2} + \frac{1}{3} = 1.83 > 1.6$

```c
#include <stdio.h>

int main()
{
    float n;
    int i;
    float sum = 0;
    printf("Enter n :");
    scanf("%f", &n);

    i = 1;
    while (sum < n) {
        sum = sum + 1.0 / i;
        i = i + 1;
        printf("%d / %d = \n", i - 1)
    }
    return 0;
}
```

\[ \frac{1}{1} + \frac{1}{2} + \frac{1}{3} \]