CS 1713
Introduction to Computer Programming II
Midterm 2

<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
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</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. (20 pts) What is the output of the following program? Show all your work

```c
#include <stdio.h>

int function1(int data[10], int a, int b)
{
    return(data[b]-data[a-1]);
}

int main()
{
    int a[10];
    int i=1,x;
    a[0]=0;
    while (i<10)
    {
        a[i] = i+a[i-1];
        i++;
    }
    for (i=3; i<7; i++)
    {
        x = function1(a,i,i+2);
        printf("%d    %d\n",i,x);
    }
}
```
2. (20 pts) What is the output of the following program? Show all your work.

```c
#include <stdio.h>

int function1(int *m)
{
    *m = *m + 1;
    return(*m);
}

int function2(int n)
{
    n = n + 2;
    return(n);
}

int main()
{
    int x = 2, y = 4;
    int *ptr = &x;
    printf("Output 1: X = %d, Y = %d\n",x,y);
    x = function2(y);
    printf("Output 2: X = %d, Y = %d\n",x,y);

    x = 2; y = 4;
    y = function1(ptr);
    printf("Output 3: X = %d, Y = %d\n",x,y);

    x = 2; y = 4;
    x = function2(function1(ptr));
    printf("Output 4: X = %d, Y = %d\n",x,y);

    x = 2; y = 4;
    x = function2(function2(y));
    printf("Output 5: X = %d, Y = %d\n",x,y);

    return 0;
}
```

Output 1: X = 2, Y = 4
Output 2: X = 10, Y = 4
Output 3: X = 3, Y = 3
Output 4: X = 5, Y = 4
Output 5: X = 8, Y = 4
3. (20 pts) C language does not provide a standard function that removing trailing spaces at the end of a string. Write a function to do this. Sample executions of the function and the function prototype are given below. You can use `strlen()` function in your implementation.

given below.

"Apple\_" becomes "Apple"
"An apple" becomes "An apple"
"Apple 2" becomes "Apple 2"

```c
char *strtrim(char *str)
{
    char *end;
    end = str + strlen(str) - 1;
    while (end > str && isspace(*end))
        end--;
    *(end+1) = '\0';
    return(str);
}
```
4. (20 pts) Consider a coin system where there are coins for 1 cent, 4 cents and 16 cents instead of the 1 cent, 5 cents, 10 cents and 25 cents that we have now. Find out whether optimal change (minimum number of coins) for every amount from 1 cent to 60 cents can be given using at most 3 coins of each type in this system. Verify this using your program or print counterexample you find if there are any. Write a Complete program to do this. You can use functions in your program if you want.
5. (20 pts) Complete the following program to find the first non-repeating character in a string. Read a string from the user and print the first non-repeating character. Hint: first non-repeating character is one of the characters in the string and appears in the string only once.

"appleforapple" has first non-repeating character 'f', since 'a','p','l','e' repeat
"an apple" has first non-repeating character 'n' since 'a' repeats
"abcddcb" has first non-repeating character 'a' since 'a' is first character and does not repeat

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

// returns how many times ch appears in str
int countchar(char ch, char *str)
{
    int count = 0;
    while (*str != '\0') {
        if (*str == ch)
            count++;
        str++;
    }
    return(count);
}

int main() {
    char str[100];
    char *str1;
    char norepeat;
    int done = 0;

    fgets(str, 100, stdin);
    str1=str;

    printf("First nonrepeating character is %c\n", norepeat);
    return 0;
}
```
CS 1713
Introduction to Computer Programming II
Midterm 2

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

NAME:__________________________

Instructions
1. Do all of the 4 problems
3. You have 50 minutes for the exam
4. Show all your work
5. Do not separate midterm papers

<table>
<thead>
<tr>
<th>Easy</th>
<th>Difficulty Level</th>
<th>Difficult</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. (20 pts) What is the output of the following program? Show all your work. Draw the contents of the array.

```c
#include <stdio.h>

int function1(int data[8], int a, int b)
{
    data[a]=b;
    return(b-1);
}

int main()
{
    int data[10];
    int i,x;

    for (i=0; i<10; i++)
        data[i]=i;

    i=2;
    while (i<=8)
    {
        x = function1(data,i,i*(i-1));
        printf("%d  %d\n",i,x);
        i = i + 1;
    }
}
```
2. (30 pts) Write a function to find the number of occurrences of "the" in a string. Count both "The" and "the" in a string. Count words that start with "the" and "The" as well. Sample execution of the function and function prototype is given below.

"The red car hit the blue car" returns 2
"An apple a day keeps the doctor away" returns 1
"There is a car on the street" returns 2

```c
int countthe(char *str)
{
    int count = 0;
    while (*str != '\0' && *str+1 != '\0' && (str+2) != '\0')
        if ((*str == 'T' || *str == 't') && *(str+1) == 'h' && *(str+2) == 'e')
            count++;
    str++;

    return (count);
}
```
3. (20 pts) Write a single function to find the minimum and maximum in an array. Add your code to the below function example. Add the additional parameters you need for minimum and maximum. size is the size of array data.

```c
void findminmax(int data[], int size, int *minptr, int *maxptr)
{
    int i;
    *minptr = data[0];
    *maxptr = data[0];
    for (i = 0; i < size; i++)
    {
        if (data[i] > *maxptr)
            *maxptr = data[i];
        else if (data[i] < *minptr)
            *minptr = data[i];
    }
}
```
4. (30 pts) **Complete the following program** to find the most frequent character in a string. If there are multiple characters with the same frequency, you can print one of them. Read a string from the user and print the most frequent character. Sample executions are given below

"apple for apple" has most frequent character 'p', since 'p' appears 4 times

"an apple a day" has most frequent character 'a' since 'a' appears 4 times

"abcdedcb" has most frequent character 'd' since 'd' appears 3 times

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

// returns how many times ch appears in str
int countchar(char ch, char *str)
{
    int count = 0;
    while (*str != '\0')
    {
        if (*str == ch)
            count++;
        str++;
    }
    return(count);
}

int main() {
    char str[100];
    char *str1;
    char mostfrequent;
    int maxcount = 0, count = 0;

    fgets(str,100,stdin);
    str1=str;

    printf("Most frequent character is %c\n",mostfrequent);
    printf("It appears %d times\n",maxcount);
    return 0;
}
```
CS 1713  
Introduction to Computer Programming II  
Midterm 2

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

NAME:______________________________

Instructions  
1. Do all of the 4 problems  
3. You have 50 minutes for the exam  
4. Show all your work  
5. 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</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. (20 pts) What is the output of the following program? Show all your work. Draw the contents of the array.

```c
#include <stdio.h>

int function1(int info[], int size, int a, int b)
{
    if (a<5)
        info[a]=b-1;
    else
        info[a]=b+1;
    return(b+2);
}

int main()
{
    int info[9];
    int i,j;

    info[0]=1;
    for (i=1; i<9; i++)
        info[i]=info[i-1]+2;

    i=1;
    while (i<=9)
    {
        j = function1(info,9, i,(i-1)*(i-1));
        printf("%d  %d\n",i,j);
        i = i + 2;
    }
}
```
2. (30 pts) Write a function strcut to truncate a string after n characters. Sample executions of the function and the function prototype are given below.

```
strcut("Apple ", 5) returns "Apple"
strcut("An apple", 2) returns "An"
strcut("Apple 2", 3) returns "App"
```

Complete the function below

```
char *strcut(char *str, int n)
{
    int i;
    char *str1 = str;
    for (i = 0; i < n; i++)
        str1[ ] = '0';
    *str1 = '0';
    return (str);
}
```
3. (20 pts) What is the output of the following program? Show all your work. Draw the contents of the array and pay attention to the formatting of the output.

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

void print(int *info, int size)
{
    int i;
    for (i=0; i<size; i++)
        printf("%d ",info[i]);
    printf("\n");
    return;
}

int main()
{
    int i,*data,*dataptr;

    data = (int *)malloc(4*sizeof(int));

    for (i=0; i<4; i++)
        data[i]=3*i;  \( \text{(data+i)} = 3 \times 1 \)

    print(data,4); \( \text{output} \)
    *data = 5;
    dataptr = data;
    dataptr++;
    *dataptr = 1;

    print(data,4);

    *(data+2) = 4;
    *(dataptr+2)=2;

    print(data,4);
    free(data);
    return 0;
}
```
4. (30 pts) Write a single function to find the pair of numbers whose sum is the largest over all the pairs in the array. Add your code to the below function example. Add the additional parameters you need for the pair of numbers. size is the size of array data. Use a SINGLE LOOP (no nested loops) to find the pair of numbers. You can assume that all the array elements are distinct.

For the following array

\[2\ 7\ 4\ 3\ 8\ 5\ 6\]

7 and 8 are the pair of numbers that generate the largest sum 15.

```c
void findlargestsumpair(int data[], int size, int pair1, int pair2)
{
    int i;
    *pair1 = data[0];
    *pair2 = data[0];
    for (i = 0; i < size; i++)
    {
        if (data[i] > *pair1)
        {
            *pair2 = *pair1;
            *pair1 = data[i];
        }
        else if (data[i] > *pair2)
        {
            *pair2 = data[i];
        }
    }
    pair2 = pair1;
}
```
node *findmax(node *list1)
{
    node *maxptr;
    if (list1 == NULL)
        return (NULL);

    maxptr = list1;
    while (list1 != NULL)
    {
        if (list1->info > maxptr->info)
            maxptr = list1;
        list1 = list1->next;
    }

    return (maxptr);
}

maxptr = findmax(head1);

if (findmax(head1) == head1)