<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>Question 6</td>
<td></td>
</tr>
<tr>
<td>Question 7</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
</tr>
</tbody>
</table>

NAME:_________________________

**Instructions**

1. Do **ONLY** 6 of the 7 problems
2. You have 90 minutes for the exam
3. Show all your work
4. Do not separate exam papers

<table>
<thead>
<tr>
<th>Easy</th>
<th>Difficulty Level</th>
<th>Difficult</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2 3</td>
<td>4 5 6 7 8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9 10</td>
</tr>
</tbody>
</table>
1. (20 pts) What is the output of the following program? Show what is printed when `printlist()` functions are called. Show all your work.

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

struct node
{
    int info;
    struct node *next;
};
typedef struct node node;

void printlist (node *head)
{
    while (head != NULL)
    {
        printf("%d ",head->info);
        head = head->next;
    }
    printf("\n");
}

int main()
{
    node *head,*ptr,*ptr2;

    head = (node*)malloc(sizeof(node));
    head->next = (node*)malloc(sizeof(node));
    head->info = 12;
    head->next->info = 9;
    head->next->next = NULL;
    printlist(head->next);
    ptr = head->next;
    printlist(head);

    ptr2 = (node*)malloc(sizeof(node));
    ptr2 -> next = ptr;
    ptr2->info = 17;
    head -> next = ptr2;
    printlist(ptr2);
    printlist(head);
}
```
2. (20 pts) Write a function to find a peak element in an array. A peak element is an element that is greater than its neighbors. Given an input array where \( num[i] \neq num[i + 1] \), find a peak element and return its index. The array may contain multiple peaks, in that case return the index to any one of the peaks. You may assume that \( num[-1] = -\infty \) and \( num[n] = -\infty \). So, \( num[0] \) and \( num[n - 1] \) could be peaks as well. For example, in array [1, 2, 3, 1], 3 is a peak element and your function should return the index number 2.

In below function prototype, \( n \) is the size of the array

```c
int peak(int num[], int n)
{
    // Your code here
}
```
3. (20 pts) Consider the following declaration of array of pointers

```c
int *data1[8];
int *data2[12];
```

Write a program fragment (for and if statements) to find out if there are two pointers in array `data1` and `data2` that point to the same location. If such pointers exist write the indexes of the arrays that point to the same location. Sample output is given below.

`data1[3] and data2[5] point to the same location`
4. (20 pts) What is the output of the following program. Show all your work.

```c
#include <stdio.h>

struct triangle
{
    float width;
    float height;
};

typedef struct triangle triangle;

int main()
{
    triangle t1={3,4};
    triangle t2={1,2};
    triangle t3={4,6};
    triangle *tptr1=&t2;
    triangle *tptr2=&t1;

    printf("1 %f %f\n",t2.width, t2.height);
    printf("2 %f %f\n",tptr2->width, tptr2->height);

    tptr1 = &t3;
    printf("3 %f %f\n",(*tptr1).width, (*tptr1).height);
    tptr2->width = 5;
    tptr1 = tptr2;

    printf("4 %f %f\n",tptr1->width, tptr1->height);
    t1 = t2;
    printf("5 %f %f\n",t1.width, t1.height);
}
```
5. (20 pts) Write a function to compute if a substring from positions i to j of a string consists of unique characters. If all the characters are different the function returns 1, otherwise it returns 0. Prototype of the function is given below.

```
int isunique(char *str, int i, int j)
{
}
```
6. (20 pts) Span of a linked list is the difference between the largest and smallest elements in the linked list. Write a function \textit{span()} to compute the span of a linked list. Node declaration of the linked list is given below.

\begin{verbatim}
struct node
{
    int info;
    struct node *next;
};
typedef struct node node;
\end{verbatim}

Span of the following list is 12-5=7.

\begin{center}
\textbf{head} -> 5 -> 6 -> 12 -> Null
\end{center}

\begin{verbatim}
int span(node *ptr)
{
}
\end{verbatim}
7. (20 pts) What is the output of the following program? Show all your work.

```
#include <stdio.h>

int function1(int a, int b)
{
    if (a % b == 2)
        return a;
    else
        return (function1(a+b,a-b));
}

int main()
{
    int x,y;

    x = 7; y = 2;
    printf("Out1 = %d\n",function1(7,2));

    x = 8; y = 3;
    printf("Out1 = %d\n",function1(8,3));

    x = 11; y = 5;
    printf("Out1 = %d\n",function1(11,5));
}
```