CMPS 221 Sample Midterm 2

Each question is worth 5 points. 50 points in total.

1. What is a call-by-value parameter and a call-by-reference parameter? How do they differ?
2. Assume you have the character 'c' declared. What is the difference between calling cin.get(c) and cin >> c? When answering, consider what would happen with each input statement when the user hits just the enter key.
3. Why is it important to make sure that a valid index number is used when accessing an array element? When answering, consider what would happen when an invalid index is used.
4. Write a void function called get_input that has two call-by-reference integer parameters. The function will prompt the user for two integers and read the integers in to the parameters. Give both the function prototype and function body. Do NOT give main.
5. Write a calculation function called find_area that will return a double for the area of a rectangle. The function will take two double call-by-value arguments for the lengths of the sides of the rectangle. This is a pure calculation function. There should be NO cin or cout statements inside of your function. Give both the function prototype and function body. Do NOT give main.
6. Write the main function which will declare an array of doubles of size 5, prompt the user to input all 5 doubles and calculates the sum of all 5 doubles.
7. Are the following function prototypes properly overloaded? If not, indicate which prototypes are in conflict.

```
int printMin(int);
void printMin(int);
void printMin(double);
void printMin(int, double);
void printMin(int=0, double=0.0);
```

8. You currently have two integer pointers called p1 and p2 and two integer variables called v1 and v2 that have the following values:

```
Address:    80000     80004     80008     80012     80016     80020     80024
           ----------------------------------------------------------------------
          |         |         |         |         |         |         |         |
Value:    |    0    |    0    |    0    |    0    |   58    |    0    |   17    |
          |_________|_________|_________|_________|_________|_________|_________|

Variable:     p1                  p2                  v1                  v2
```

Show the effect of executing the following code:

```
  p1 = &v1;
  p2 = &v2;
  p2 = p1;
  *p1 = 30;
```

```
Address:    80000     80004     80008     80012     80016     80020     80024
           ----------------------------------------------------------------------
          |         |         |         |         |         |         |         |
Value:    |         |         |         |         |         |         |         |
          |_________|_________|_________|_________|_________|_________|_________|

Variable:     p1                  p2                  v1                  v2
```

9. What is the output of the following program?

```
#include <iostream>
#include <stdio.h>
using namespace std;

void rearrange(int &, int, int, int &);

int main()
{
  int num1 = 2, num2 = 5, num3 = 8, num4 = 10;

  printf("Before calling function:\n");
  printf("num1=%d num2=%d num3=%d num4=%d\n", num1, num2, num3, num4);

  rearrange(num1, num2, num3, num4);

  printf("After calling function:\n");
  printf("num1=%d num2=%d num3=%d num4=%d\n", num1, num2, num3, num4);

  return 0;
}

void rearrange(int &n1, int n2, int n3, int &n4)
{
  int tmp1 = n1, tmp2 = n3;

  n1 = n2;
  n2 = tmp2;
  n3 = n4;
  n4 = tmp1;
}
```

10. There are 5 syntax and logical errors in this program. List the errors and how to fix them.

```
#include <iostream>
using namespace std;

void average(int *, int, double);

int main()
{
  int size = 5, avg;
  int arr[5];
  arr = {4, 2, 8, 3, 7};

  printf("The fifth element is %d\n", a[5]);

  average(arr, size, avg);
  printf("The average is %d, avg.\n");

  return 0;
}

void average(int *a, int size, double avg)
{
  int sum = 0, i;

  for(i = 1; i < size; i++)
  {
    sum += a[i];
  }

  avg = sum / size;
}
```