1.[a]

```
int main ()
{
      int arr[3] = {5,10,15};
      int* ptr = arr;

      *ptr = 30;          //set arr[0] to 30
      *(ptr + 1) = 20;    //set arr[1] to 20
      ptr += 2;
      ptr[0] = 10;        //set arr[2] to 10
      ptr -= 2;

      while (ptr < arr + 3)
      {
            cout << *ptr << endl;     // print values
            ptr++;
      }
}
```

1.[b]

This function can't work because the pToMax parameter is not passing by reference. So when pToMax change in the function findMax, the ptr will not change with it in the main function.

```
  void findMax(int arr[], int n, int*& pToMax)
    {
        if (n <= 0)
            return;       // no items, no maximum!

        pToMax = arr;

        for (int i = 1; i < n; i++)
        {
            if (arr[i] > *pToMax)
                 pToMax = arr + i;
        }
    }

    int main()
    {
        int nums[4] = { 5, 3, 15, 6 };
        int* ptr;

        findMax(nums, 4, ptr);
        cout << "The maximum is at address " << ptr << endl;
        cout << "It's at position " << ptr - nums << endl;
```

```
        cout << "Its value is " << *ptr << endl;
    }
```

1.[c]

The problem that the main function has is that the pointer ptr it gave to function computeCube is a null pointer.

```
  void computeCube(int n, int* ncubed)
    {
     *ncubed = n * n * n;
    }

    int main()
    {
     int a;
     int* ptr = &a;
     computeCube(5, ptr);
     cout << "Five cubed is " << *ptr << endl;
    }
```

1.[d]

The problems with the implementation of the function are using str1, str2 instead of using *str1, *str2. str1, str2 are names of C strings char str1[] and char str2[] but not pointers to them. Only *str1 and *str2 represent the pointers of str1[] and str2[].

```
      // return true if two C strings are equal
    bool strequal(const char str1[], const char str2[])
    {
        while (*str1 != 0  &&  *str2 != 0)
        {
            if (*str1 != *str2)  // compare corresponding characters
                return false;
            str1++;            // advance to the next character
            str2++;
        }
        return *str1 == *str2;   // both ended at same time?
    }

    int main()
    {
        char a[15] = "Noor";
        char b[15] = "Noah";

        if (strequal(a,b))
            cout << "They're the same person!\n";
```

```
    }
```

1.[e]

Function getPtrToArray returns an address of an array that does not exist in the main function.

2.

```
[a]double* cat;
[b]double mouse[5];
[c]cat = &mouse[4];
[d]*cat = 42;
[e]*(mouse+3) = 25;
[f]cat -= 3;
[g]cat[1] = 17;
[h]cat[0] = 54;
[i]bool b;
   if (*cat == *(cat+1))
     b = true;
   else
     b = false;
[j]bool d;
   if (*cat == mouse[0] )
     d = true;
   else
     d = false;
```

3.[a]

```
   double mean(const double* scores, int numScores)
    {
        double tot = 0;
        int i = 0;
        while ( i < numScores)
        {
            tot += *(scores+i);
            i++;
        }
        return tot/numScores;
    }
```

3.[b]

```
  // This function searches through str for the character chr.
    // If the chr is found, it returns a pointer into str where
    // the character was first found, otherwise nullptr (not found).

    const char* findTheChar(const char* str, char chr)
    {
```

```
        for (int k = 0; *(str+k) != 0; k++)
        {
               if (*(str+k) == chr)
                  return str+k;
         }
         return nullptr;
    }
```

3.[c]

```
const char* findTheChar(const char* str, char chr)
    {
        while(*str != 0)
        {
           if (*str == chr)
                 return str;
           str += 1;
         }
        return nullptr;
    }
```

4.

The program prints:

```
3
4
79
-1
9
22
19
```

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

    int* maxwell(int* a, int* b)
    {
        if (*a > *b)//array[0] = 5, array[2] = 4, 5 > 4
            return a;//return array[0]
        else
            return b;
    }

    void swap1(int* a, int* b)
    {
        int* temp = a;//temp = &array[0]
        a = b;//&array[0] = &array[1]
        b = temp;//&array[1] = temp
    }
```

```
void swap2(int* a, int* b)
{
    int temp = *a;//temp = array[0] = -1
    *a = *b;//array[0] = array[2] = 4
    *b = temp;//array[2] = temp = -1
}

int main()
{
    int array[6] = { 5, 3, 4, 17, 22, 19 };

    int* ptr = maxwell(array, &array[2]);
   *ptr = -1;//Assigned array[0] to -1.
    ptr += 2;//point to array[2]
    ptr[1] = 9;//set array[3] to 9
    *(array+1) = 79;//set array[1] to 79

    cout << &array[5] - ptr << endl;//ptr = &array[2], print out 3

    swap1(&array[0], &array[1]);//no effect
    swap2(array, &array[2]);

    for (int i = 0; i < 6; i++)
        cout << array[i] << endl;//print out {4,79,-1,9,22,19}
}
```

5.

```
void removeS(char* msg)
{
    char* ptr;
    while (*msg != 0)
    {
        if (*msg == 's' || *msg == 'S')
        {
            ptr = msg;
            for (;*msg != 0; msg++)
            {
                *msg = *(msg + 1);
            }
            msg = ptr;
            continue;
        }
        msg++;
    }
}
```

```
int main()
{
    char msg[50] = "She'll be a massless princess.";
    removeS(msg);
    cout << msg;  // prints   he'll be a male prince.
}
```