Problem 1 (20 points): Short Answer

Part A (5 points): Give one reason for using local variables instead of global variables.

Part B (5 points): Describe one advantage of using a debugger.

Part C (5 points): Write the output of the following C code.

```
for( i = 0 ; i < 4 ; i++) {
      if( i == 2 )
            continue;
      for( j = 0 ; j < 4 ; j++){
            if( j == 2 )
                  break;
            printf( "%d %d\n" , i , j );
      }
}
```

Part D (5 points): You and a friend are charged with writing a function that reorganizes an array of integers by discarding any negative elements and adjusting the count of valid elements. Your friend proposes the following declaration:

```
void keep_positive (int* array, int num_things);
```

Explain why this declaration cannot serve the intended purpose.

Part C (4 points): The following is a code fragment from a C source file.

```
int a;

int foo()
{
      int* b = &a;
}
```

In what part of the program memory (e.g., system memory, code, etc.) are the following values stored?

   i) `a`

   ii) `b`

   iii) `&b`

Problem 5 (20 Points): Programming in C

Part A (15 points): Write a function that returns a pointer to the **last** occurrence of the character `c` in the string `s`. If `s` does not contain `c`, your function should return NULL.

```
char* last_occurrence (char c, char* s)
{




}
```

***Part B (5 points): Write a function that returns a pointer to the **last** occurrence of the **string** `sub` in the string `s`. If `s` does not contain `sub`, your function should return NULL.

```
char* last_substring (char* sub, char* s)
{




}
```