1. (5 points) What is a template in C++? Describe an example when a C++ template would be useful?

2. (5 points) Describe the functions cin.peek(), cin.ignore(), and isspace().

3. (5 points) Describe how to use GDB to find the location of a Segmentation Fault.

4. (10 points) What are the similarities and differences between a stack and a queue? Give an example of a problem solved naturally using a stack as well as one using a queue.
5. (10 points) What does the following program print. Show the steps of the calculation for ANY credit.

Be very methodical, it is easy to make a mistake.

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

int f(int n[], int size)
{
    if (size <= 1)
        return 1;
    return (n[0] + n[size-1]) * f(n+1, size-2);
}

int main()
{
    int nums[] = {1,2,3,4}; // initializes array so first element is 1, 2nd is 2, etc
    cout << f(nums, 4) << endl;
}
```

6. (10 points) Write the recursive function product(). This function recursively calculates the product of all the elements in an array. Assume the given array has at least 1 element. Think carefully about the base case.

```cpp
int product(int array[], int size)
{
}
```
7. (10 points) Using the class Pqueue definition on the last page, write `~Pqueue()`. 

Pqueue::~Pqueue()
{
}

8. (10 points) Using the class Pqueue definition on the last page, write `dequeue()`. Return a pointer to the Cust object at the front of the queue. If the queue is empty, return NULL. Make sure your solution works with the `enqueue()` you write in the next question.

Cust *Pqueue::dequeue()
{
}

9. (15 points) Using the class Pqueue definition on the last page, write the function `enqueue( Cust *cust, int priority)` which enqueues the given Cust object onto the queue using the given priority. The new element should be inserted so that this function works with `dequeue()` in the previous question.

void Pqueue::enqueue(int priority, Cust *cust)
{
}