The C++ Memory Model: Clicker

```cpp
int main(){
    float num_cakes = 10;
    float[] number_array = {1,2,3};
    std::cout << "Half the cakes is " << half(num_cakes) << " cakes." << std::endl;//Line A
    float num_pies = 3;//Line C
}
float half(float input){
    float quotient = input * 0.5f;
    return quotient;
}
```

What does the stack look like at Line A?

**Stack 1**
- float num_cakes: 5.0f
- float[] num_array:
  - Index 0: 1.0f
  - Index 1: 2.0f
  - Index 2: 3.0f
- float num_cakes: 5.0f

**Stack 2**
- float[] num_array:
  - Index 0: 1.0f
  - Index 1: 2.0f
  - Index 2: 3.0f
- float num_cakes: 5.0f
- float quotient: 5.0f

**Stack 3**
- float num_cakes: 5.0f
- float[] num_array:
  - Index 0: 1.0f
  - Index 1: 2.0f
  - Index 2: 3.0f
- float quotient: 5.0f
The C++ Memory Model: Clicker

What does the stack look like at Line A?
ANSWER: Stack 1 (stack is bottom-up)
The C++ Memory Model: Clicker

Which line of execution does this stack represent?

1. Line A
2. Line B
3. Line C
### The C++ Memory Model: Clicker

```cpp
int main()
{
    float num_cakes = 10;
    float[] number_array = {1,2,3};

    std::cout << "Half the cakes is " << half(num_cakes) << " cakes." << std::endl; // Line A
    float num_pies = 3; // Line C

    float half(float input){
        float quotient = input * 0.5f;
        return quotient;
    }
}
```

Which line of execution does this stack represent?

1. Line A
2. Line B
3. Line C

ANSWER: Line B

<table>
<thead>
<tr>
<th>Index</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0f</td>
</tr>
<tr>
<td>1</td>
<td>2.0f</td>
</tr>
<tr>
<td>2</td>
<td>3.0f</td>
</tr>
</tbody>
</table>

| float num_cakes | 5.0f |
Pointers vs References: Clicker

Which memory model is correct?
Pointers vs References: Clicker

```
int a = 5;
int* ptr = &a;
int& ref = a;
(*ptr) = 6;
ref = 7;
ptr = 8;
```

<table>
<thead>
<tr>
<th></th>
<th>STACK</th>
<th>STACK</th>
<th>STACK</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>int a</td>
<td>6</td>
<td>int a</td>
</tr>
<tr>
<td></td>
<td>int* ptr</td>
<td>8</td>
<td>int* ptr</td>
</tr>
<tr>
<td></td>
<td>int&amp; ref</td>
<td>7</td>
<td>int&amp; ref</td>
</tr>
</tbody>
</table>

ANSWER: 2 (technically, this wouldn’t compile, though)
The C++ Memory Model: Clicker

```cpp
int main()
{
    Node* root = new Node();
    Node* n2 = new Node(root);
}

class Node{
    public:
        Node* child;
        Node* parent;
    Node(Node* p){
        parent = p;
        p->child = this;
    }
};
```

Does this memory model represent the code to the left?
1. Yes
2. No
The C++ Memory Model: Clicker

int main()
{
    Node* root = new Node();
    Node* n2 = new Node(root);
}

class Node{
public:
    Node* child;
    Node* parent;

    Node(Node* p){
        parent = p;
        p->child = this;
    }
};

Does this memory model represent the code to the left?
1. Yes
2. No
ANSWER: Yes
The C++ Memory Model: Clicker

Q1: Which vec4 initialization is incorrect?
1. v1
2. v2
3. v3
4. v4
5. v5

Q2: Which of the vec4s are null?
1. v3
2. v4
3. v5
4. None of the above
Q1: Which vec4 initialization is incorrect?
1. v1
2. v2
3. v3
4. v4
5. v5
ANSWER: v2

Q2: Which of the vec4s are null?
1. v3
2. v4
3. v5
4. None of the above
ANSWER: None of the above
Destructors: Clicker

Which destructor is the best implementation?
1. Destructor A
2. Destructor B
3. Destructor C
4. They’re all terrible

Which of them would cause a memory leak?
1. Destructor A
2. Destructor B
3. Destructor C
4. None
Destructors: Clicker

Which destructor is the best implementation?
1. Destructor A
2. Destructor B
3. Destructor C
4. They’re all terrible

ANSWER: C

Which of them would cause a memory leak?
1. Destructor A
2. Destructor B
3. Destructor C
4. None

ANSWER: NONE, but A would cause a crash
std::vectors: Clicker

What are the final contents of array1?

1. A
2. B
3. C
4. D
std::vectors: Clicker

What are the final contents of array1?

1. A
2. B
3. C
4. D

ANSWER: B
Traversal: Clicker

What does this scene graph produce?

Assume a transformation order of: Translate * Rotate * Scale * Geometry
Traversal: Clicker

What does this scene graph produce?

Assume a transformation order of:
Translate * Rotate * Scale * Geometry

ANSWER: Option 1
Clicker Round 2

What does this scene graph produce?

Assume a transformation order of:
Translate * Rotate * Scale * Geometry
Clicker Round 2

What does this scene graph produce?

Assume a transformation order of:
Translate * Rotate * Scale * Geometry

ANSWER: Option 2
VBOs: Clicker

How long will our position, normal, and index arrays have to be for this cube?

1. Pos: 8, Nor: 24, Idx: 24
2. Pos: 8, Nor: 8, Idx: 24
3. Pos: 8, Nor: 24, Idx: 36
4. Pos: 8, Nor: 36, Idx: 36
5. Pos: 24, Nor: 24, Idx: 24
6. Pos: 24, Nor: 36, Idx: 36
7. Pos: 24, Nor: 24, Idx: 36
VBOs: Clicker

How long will our position, normal, and index arrays have to be for this cube?

1. Pos: 8, Nor: 24, Idx: 24
2. Pos: 8, Nor: 8, Idx: 24
3. Pos: 8, Nor: 24, Idx: 36
4. Pos: 8, Nor: 36, Idx: 36
5. Pos: 24, Nor: 24, Idx: 24
6. Pos: 24, Nor: 36, Idx: 36
7. Pos: 24, Nor: 24, Idx: 36

ANSWER: 7