1 Preliminaries (1 minute, 1 point)

A. Print your name on your exam booklet.
B. Print your netid next to your name.
C. `printf("Precept %s", your_precept_number)` on your exam booklet (refer to the table below).
D. Write and sign the honor pledge.

Precept sections:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MW 1:30</td>
<td>Petras</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>MW 3:30</td>
<td>Petras</td>
<td>5A</td>
</tr>
<tr>
<td>3</td>
<td>MW 7:30</td>
<td>Austin Le</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>TTh 12:30</td>
<td>Xiaoyan Li</td>
<td>6A</td>
</tr>
<tr>
<td>4A</td>
<td>TTh 12:30</td>
<td>Gabai</td>
<td>7</td>
</tr>
</tbody>
</table>

2 Numbers (8 minutes, 10 points)

Show the output of this program. Remember that `%x` is a format indicator to print in hexadecimal. Use uppercase or lowercase in your hexadecimal, we don’t care.

```c
#include <stdlib.h>
#include <stdio.h>

int main(void) {
    printf("A. %x\n", 1<<5);
    printf("B. %x\n", (3<<3)|(3<<2));
    printf("C. %x\n", (-1)&15);
    printf("D. %x\n", 5|7);
    printf("E. %x\n", 4-7);

    return EXIT_SUCCESS;
}
```
3 Modular programming (25 minutes, 25 points) start writing on a left-hand page

A vector is a sequence of integers. This program manipulates values of type vector, that is `Vec_T`:

```c
/* this is vclient.c */
#include <stdlib.h>
#include <stdio.h>
#include "vector.h"

int main(void) {
    Vec_T u, v, w;
    v = Vec_iota(5); /* v = [1,2,3,4,5] */
    u = Vec_offset(4, v); /* u = [5,6,7,8,9] */
    Vec_free(v);
    w = Vec_truncate(3, u); /* w = [5,6,7] */
    v = Vec_offset(-1, w); /* v = [4,5,6] */
    printf("%d\n", Vec_first(v)); /* prints 4 */
    Vec_free(u);
    Vec_free(v);
    Vec_free(w);
    return EXIT_SUCCESS;
}
```

Your job is to write `vector.h` and `vector.c` that implements the abstract data type `Vec_T`. Use a representation that looks like this,

```
3
5
6
7
```
represents the value [5,6,7]

To keep things simple,
- Support only the operations used here by vclient.c.
- Assume `malloc` always succeeds (never returns `NULL`).
- Don’t write any comments or asserts. Don’t check for errors.

Note: `w=Vec_truncate(3, u)` creates a new vector `w` with copies of the first three elements of `u`, without modifying vector `u`. Similarly, `Vec_offset` creates a new vector without disturbing the old one.

4 Change of representation (5 minutes, 5 points)

Now suppose we want to use a representation that looks like this:

```
3
5
6
7
```
represents the value [5,6,7]

Show what `vector.h` would look like in this case. DO NOT show `vector.c`.

5 Unit test (9 minutes, 9 points)

A. In 25 words or less, explain why the boneheaded implementation at right is not appropriate.

B. Write a concise unit test for this module, `testvector.c`, just to detect whether the boneheaded implementation was used. No output on correct implementation, assertion failure on bonehead.

6 Take home problem (0 points, optional)

What’s the most efficient representation of the vector type that’s compatible with your header file? Discuss with your friends, implement it …