CS330: Operating Systems
Quiz#1

Name:
Roll No.:

1. Consider the following program.

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <stdlib.h>

typedef struct {
   int x;
} global_t;

global_t *g;

int f (int n)
{
        if (n==1) return 1;
        if (fork() == 0) {
           g->x++;
           return n*f(n-1);
        }
        else {
           wait(NULL);
           return g->x;
        }
}

int main(void)
{
        int y;
        g = (global_t*)malloc(sizeof(global_t));
        g->x = 5;
        y = f(g->x-2);
        printf("%d\n", y*g->x);
        return 0;
}
```

What are the possible outputs of this program? Assume that all `fork()` calls work as expected. **(6 points)**

Solution: The output is shown below.

```
42
108
25
```

Grading policy: 2 marks per correct value in the correct order. 1 mark per correct value otherwise.

2. Consider the following programs.

```
#include <stdio.h>

int x = 0;

int main (void) {
   int fd[2];

   pipe(fd);
   if (fork() == 0) {
      x++;
      close(fd[1]);
      printf("%d\n", x);
      read(fd[0], &x, sizeof(int));
      close(fd[0]);
      printf("%d\n", x);
      x++;
   }
   else {
      x--;
      close(fd[0]);
      printf("%d\n", x);
      write(fd[1], &x, sizeof(int));
      close(fd[1]);
      wait();
      printf("%d\n", x);
   }

   x++;

   return 0;
}
```

What are the possible outputs? **(2 points)**

Solution: If the parent continues to run after fork() till the wait() call, the output will be as follows.

```
-1
1
-1
-1
```

If the child is scheduled after fork() and it runs till read(), at which point the parent is scheduled and continues till wait(), the output will be as follows.

```
1
-1
```

```
-1
-1
```

There is no other possibility.
Grading policy: 2 marks if both outputs are provided. 1.5 marks if any one output is provided. 0.5 marks if some numbers are correct, but the overall output is wrong.

3. Consider the following program.

```
#include <stdio.h>

int x = 0;

int main (void)
{
   if (fork() == 0) {
      x++;
      execv("prog1", NULL);
      printf("%d\n", x);
   }
   else {
      if (fork() == 0) {
         x++;
         sleep(10);
         execv("prog", NULL);
         printf("%d\n", x);
      }
      else {
         wait(NULL);
         printf("%d\n", x);
         wait(NULL);
         printf("%d\n", x);
      }
   }

   return 0;
}
```

The executable `prog1` does not exist. The program `prog.c` is shown below.

```
#include <stdio.h>

int main (void)
{
   sleep(1);
   return 0;
}
```

What are the possible outputs? **(2 points)**

Soution: The call to execv with `prog1` as argument will return back to the calling function. The second child does not print anything, but takes at least 11 seconds before terminating. Therefore, the print from the first child

will appear first. Then the parent will pass through the first wait() call and print. Finally, the parent will print after the second wait() call returns. The output is shown below.

```
1
0
0
```

Grading policy: 2 marks for correct output. 1.5 marks if any additional output is provided. 1 mark if numbers are correct, but appear in wrong order. 0.5 marks if some numbers are correct, but the overall output is wrong.