Process control

The next problem concerns the following four versions of the `tfgets` routine, a timeout version of the Unix `fgets` routine.

The `tfgets` routine waits for the user to type in a string and hit the return key. If the user enters the string within 5 seconds, the `tfgets` returns normally with a pointer to the string. Otherwise, the routine "times out" and returns a NULL string.

`tfgets`: Version A

```
void handler(int sig) {
  siglongjmp(env, 1);
}

char *tfgets(char *s, int size, FILE *stream) {
  pid_t pid;
  signal(SIGCHLD, handler);

  if (!sigsetjmp(env, 1)) {
    pid = fork();
    if (pid == 0) {
      return fgets(s, size, stream);
    }
    else {
      sleep(5);
      kill(pid, SIGKILL);
      wait(NULL);
      return NULL;
    }
  }
  else {
    wait(NULL);
    exit(0);
  }
}
```

`tfgets`: **Version B**

```
void handler(int sig) {
  wait(NULL);
  siglongjmp(env,1);
}

char *tfgets(char *s, int size, FILE *stream) {
  pid_t pid;

  signal(SIGUSR2, handler);
  if (sigsetjmp(env, 1) != 0)
    return NULL;
  if ((pid = fork()) == 0) {
    sleep(5);
    kill(getppid(), SIGUSR2);
    exit(0);
  }
  fgets(s, size, stream);
  kill(pid, SIGKILL);
  wait(NULL);
  return s;
}
```

`tfgets`: **Version C**

```
void handler(int sig) {
  wait(NULL);
  siglongjmp(env, 1);
}

char *
tfgets(char *s, int size, FILE *stream) {
  pid_t pid;
  str = NULL;
  signal(SIGCHLD, handler);

  if ((pid = fork()) ==  0) {
    sleep(5);
    exit(0);
  }
  else {
    if (sigsetjmp(env, 1) == 0) {
      str = fgets(s, size, stream);
      kill(pid, SIGKILL);
      pause();
    }
    return str;
  }
}
```

`tfgets`: Version D

```
void handler(int sig) {
  wait(NULL);
  siglongjmp(env, 1);
}

char *
tfgets(char *s, int size, FILE *stream) {
  pid_t pid;
  str = NULL;
  signal(SIGCHLD, handler);

  if ((pid = fork()) ==  0) {
    sleep(5);
    return NULL;
  }
  else {
    if (sigsetjmp(env, 1) == 0) {
      str = fgets(s, size, stream);
      kill(pid, SIGKILL);
      pause();
    }
    return str;
  }
}
```

Problem 60. (8 points):

This problem concerns the four versions of `tfgets` from the previous pages. Some of them are correct, and others are flawed because the author didn't understand basic concepts of concurrency and signaling. Circle the versions that are correct, in the sense that they return the input string if typed within 5 seconds, timeout after 5 seconds by returning NULL, and correctly reap their terminated children.

`Version A`   `Version B`   `Version C`   `Version D`

Note: The `pause` function sleeps until a signal is received and then returns.