CSCB09H Worksheet: Fork and Wait

For this question you will write a program that forks one child for each command line argument. The child computes the length of the command line argument and exits with that integer as the return value. The parent sums these return codes and reports the total length of all the command line arguments together. For example if your program is called `spread_the_work` and is called as `spread_the_work divide the load` it prints `The length of all the args is 13`. We have provided some parts of the code and you must work within this framework and complete the missing pieces. You do **not** need to write `include` statements.

```
int main(int argc, char **argv) {
    // Declare any variables you need
```

```
int i
pid_t pid;
```

```
    // Write the code to loop over the command line arguments.
    // (Remember to skip the executable name.)

    for ( int i=1 ; i<argc ; i++ ) {

        // call fork
        pid = fork();

        if ( pid<0 ) { // System call error

            // Handle the error
```

```
perror ("fork failed");
exit();
```

```
        } else if ( pid == 0 ) { // Child process

            // Child does work here
```

```
int len = strlen(argv[i]);
exit(length);
```

```
        }
    }


    // Parent process.
    // On the next page, finish the code to sum up
    // the return values from the child processes.
```

```
for (int i=1; i<argc; i++) {
    int status;
    wait(&status);
    if (WIFEXITED(status)) {
        sum += WEXITSTATUS(status);
    }
}
```

```
    printf("The length of all the args is %d\n", sum);
    return 0;
}
```