

EX1:

p10

p11

p2

p3

p13

p4

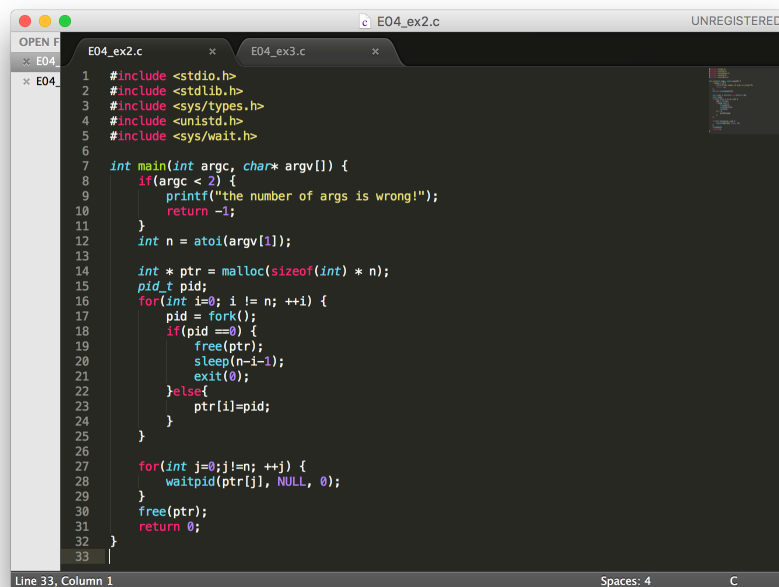
Please input a number between -128 and 127:

5

p5

p14

EX2:



The screenshot shows a code editor window titled "E04_ex2.c" with a dark theme. The code is a C program that demonstrates process management using `fork()`, `waitpid()`, `free()`, `sleep()`, and `exit()`. It includes standard headers like `<stdio.h>`, `<stdlib.h>`, `<sys/types.h>`, `<unistd.h>`, and `<sys/wait.h>`. The `main` function checks the number of arguments, converts the first argument to an integer, and then forks a series of processes. Each child process sleeps for a different duration before exiting. The parent process waits for each child to finish before freeing its memory and returning.

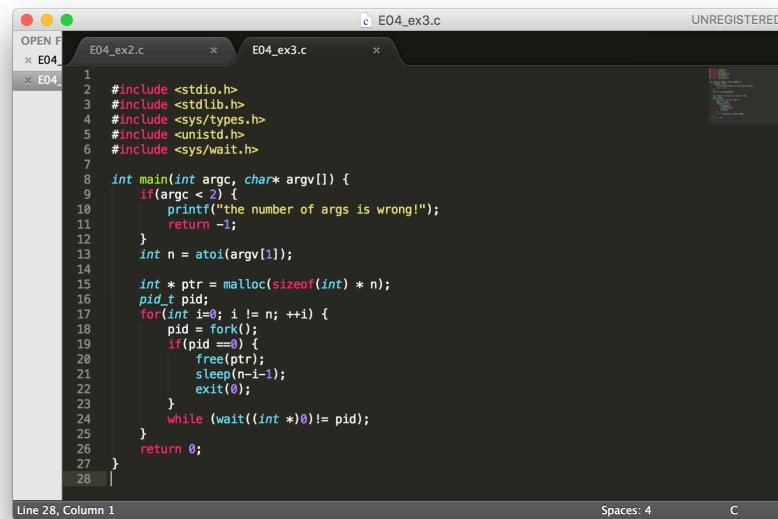
```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sys/types.h>
4 #include <unistd.h>
5 #include <sys/wait.h>
6
7 int main(int argc, char* argv[]) {
8     if(argc < 2) {
9         printf("the number of args is wrong!");
10        return -1;
11    }
12    int n = atoi(argv[1]);
13
14    int * ptr = malloc(sizeof(int) * n);
15    pid_t pid;
16    for(int i=0; i != n; ++i) {
17        pid = fork();
18        if(pid == 0) {
19            free(ptr);
20            sleep(n-i-1);
21            exit(0);
22        } else {
23            ptr[i]=pid;
24        }
25    }
26
27    for(int j=0;j!=n; ++j) {
28        waitpid(ptr[j], NULL, 0);
29    }
30    free(ptr);
31    return 0;
32 }
33
```

Line 33, Column 1 Spaces: 4 C

EX3:

Parent has to call wait() n times, the same with waitpid().

But, runs much slower.



```
1
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <sys/types.h>
5 #include <unistd.h>
6 #include <sys/wait.h>
7
8 int main(int argc, char* argv[]) {
9     if(argc < 2) {
10         printf("the number of args is wrong!");
11         return -1;
12     }
13     int n = atoi(argv[1]);
14
15     int * ptr = malloc(sizeof(int) * n);
16     pid_t pid;
17     for(int i=0; i != n; ++i) {
18         pid = fork();
19         if(pid == 0) {
20             free(ptr);
21             sleep(n-i-1);
22             exit(0);
23         }
24         while (wait((int *)0) != pid);
25     }
26     return 0;
27 }
28
```

Line 28, Column 1 Spaces: 4 C