

1.

```
main.c X
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      if(!system("dir"))
7          printf("\nmukodik\n\n");
8      system("color 2");
9      system("valami");
10
11     return 0;
12 }
13
```

```
D:\CodeBlocks\wyq5jk\bin\Debug\wyq5jk.exe
Volume in drive D is HDD
Volume Serial Number is EAF3-B0F9

Directory of D:\CodeBlocks\wyq5jk

2021. 03. 10. 23:45 <DIR> .
2021. 03. 10. 23:45 <DIR> ..
2021. 03. 10. 23:42 <DIR> bin
2021. 03. 10. 23:44 187 main.c
2021. 03. 10. 23:42 <DIR> obj
2021. 03. 10. 23:38 1 053 wyq5jk.cbp
2021. 03. 10. 23:45 106 wyq5jk.depend
3 File(s) 1 346 bytes
4 Dir(s) 243 788 042 240 bytes free

mukodik

'valami' is not recognized as an internal or external command,
operable program or batch file.

Process returned 0 (0x0) execution time : 0.087 s
Press any key to continue.
_
```

2.

```
*main.c X
1      #include <stdio.h>
2      #include <stdlib.h>
3      #include <stdbool.h>
4
5      int main()
6      {
7          char szoveg[20];
8          do{
9              scanf("%s", szoveg);
10             system(szoveg);
11         }while(true);
12
13         return 0;
14     }
15
```

3.

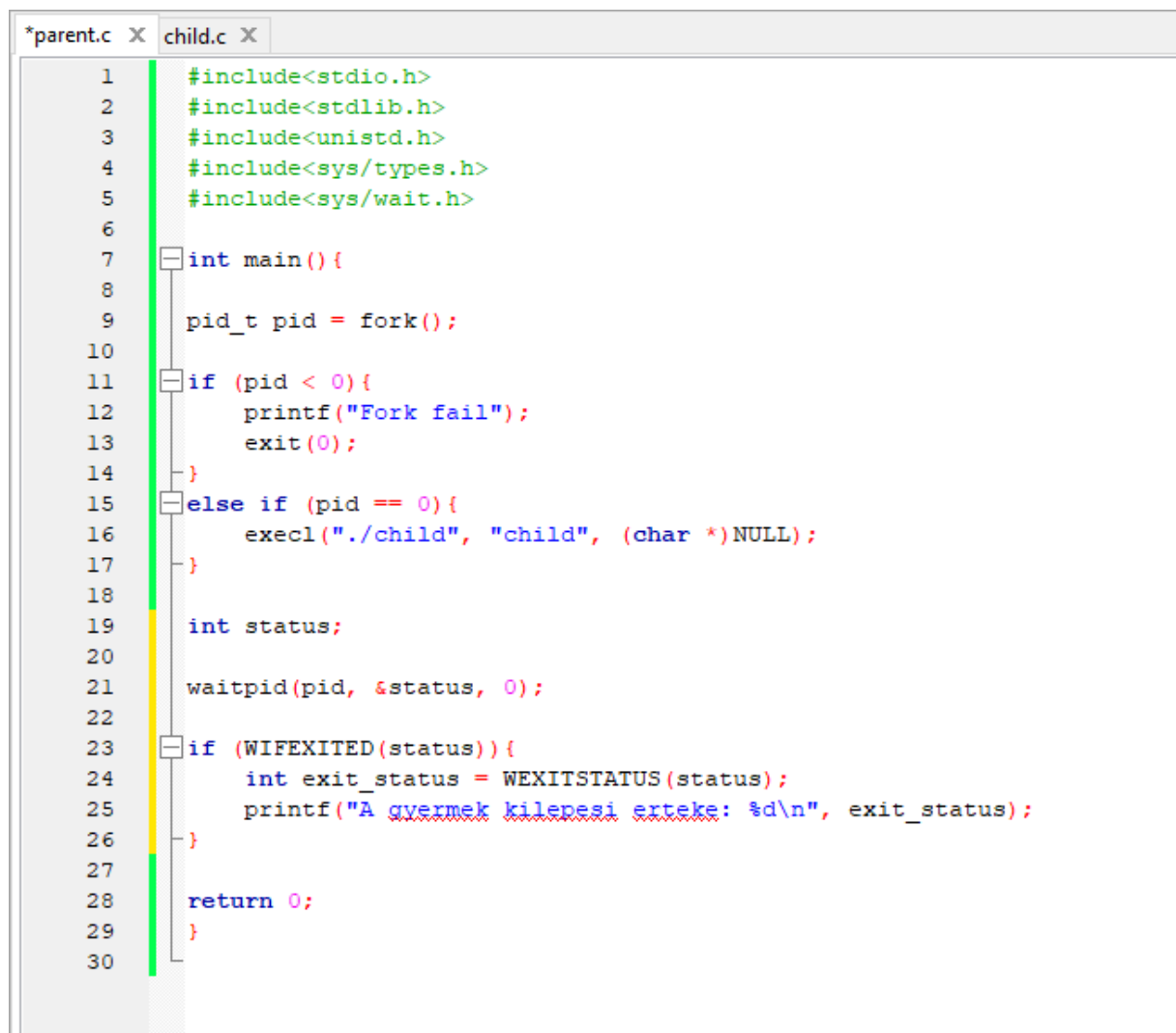
```
parent.c X child.c X
1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<unistd.h>
4  #include<sys/types.h>
5  #include<sys/wait.h>
6
7  int main() {
8
9      pid_t pid = fork();
10
11     if (pid < 0) {
12         printf("Fork fail");
13         exit(0);
14     }
15     else if (pid == 0) {
16         execl("./child", "child", (char *)NULL);
17     }
18
19     waitpid(pid, NULL, 0);
20
21     return 0;
22 }
23
```

```
parent.c X child.c X
1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<unistd.h>
4
5  int main() {
6
7      for (int i=0; i<5; i++) {
8          printf("Koloszvari Patrik --- WYQ5JK\n");
9          sleep(1);
10     }
11
12     return 0;
13 }
14
```

4.

Ugyanaz

5.



```
*parent.c x child.c x
1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<unistd.h>
4  #include<sys/types.h>
5  #include<sys/wait.h>
6
7  int main(){
8
9      pid_t pid = fork();
10
11     if (pid < 0){
12         printf("Fork fail");
13         exit(0);
14     }
15     else if (pid == 0){
16         execl("./child", "child", (char *)NULL);
17     }
18
19     int status;
20
21     waitpid(pid, &status, 0);
22
23     if (WIFEXITED(status)){
24         int exit_status = WEXITSTATUS(status);
25         printf("A gyermek kilepesi erteke: %d\n", exit_status);
26     }
27
28     return 0;
29 }
30
```

exit: a kilepesi ertekek valtozik

abort: bezarja a processzt

nullaval valo osztas: nem lehetsleges (hiba)

6.

Nem igazan ertem a feladatot. ☹