QUES:

WAP (using fork() and/or exec() commands) where parent and child execute:

a. **same program, same code**

b. **same program, different code**

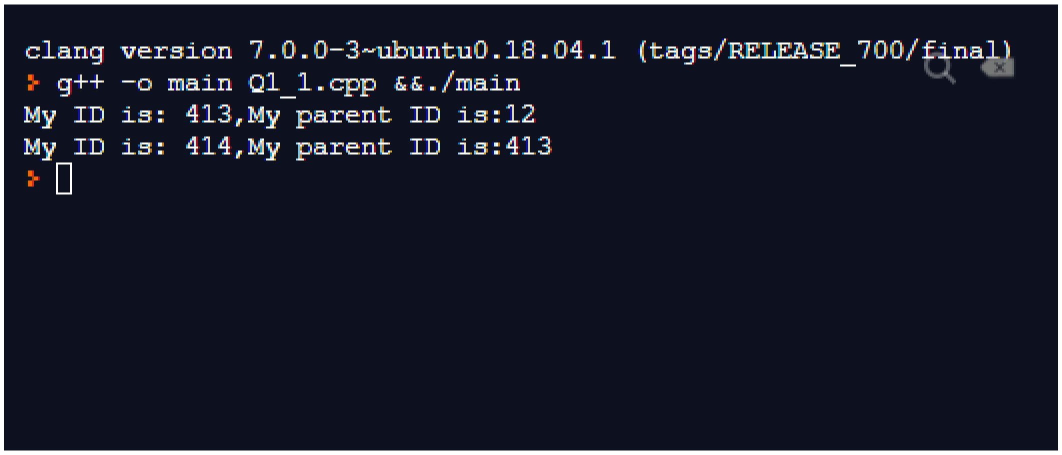
c. **different programs**

d. **before terminating, the parent waits for the child to finish its task**

1) same program, same code

```c
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/types.h>
//same program,same code
int main()
{
    int a;
    a=fork();
    if(a<0)
    {
      printf("child process could not be created");
      exit(-1);
    }
    else
    {
      printf("My ID is: %d,My parent ID is:%d\n",getpid(),getppid());
    }
    return 0;
}
```

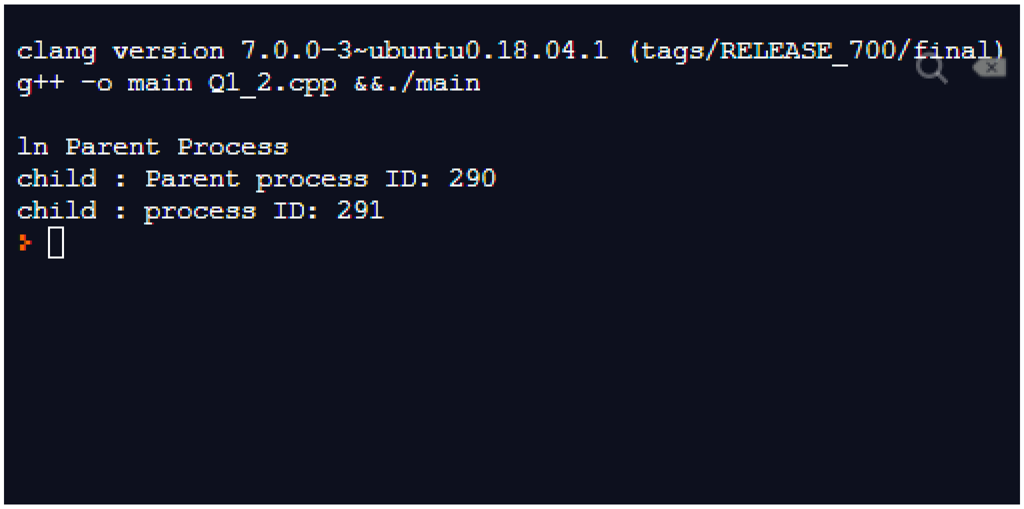
Output:


```
clang version 7.0.0-3~ubuntu0.18.04.1 (tags/RELEASE_700/final)
> g++ -o main Q1_1.cpp &&./main
My ID is: 413,My parent ID is:12
My ID is: 414,My parent ID is:413
>
```

2) same program, different code

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/types.h>
//same program,different code
int main()
{
    int a;
    a=fork();
    if(a<0)
    {
      printf("child process could not be created");
      exit(-1);
    }
    else if(a==0)
    {
      printf("child : Parent process ID: %d\n",getppid());
      printf("child : process ID: %d\n",getpid());
    }
    else{
          printf("\nln Parent Process\n");
    }
    return 0;
}
```

OUTPUT:



```
clang version 7.0.0-3~ubuntu0.18.04.1 (tags/RELEASE_700/final)
g++ -o main Q1_2.cpp &&./main

ln Parent Process
child : Parent process ID: 290
child : process ID: 291
```

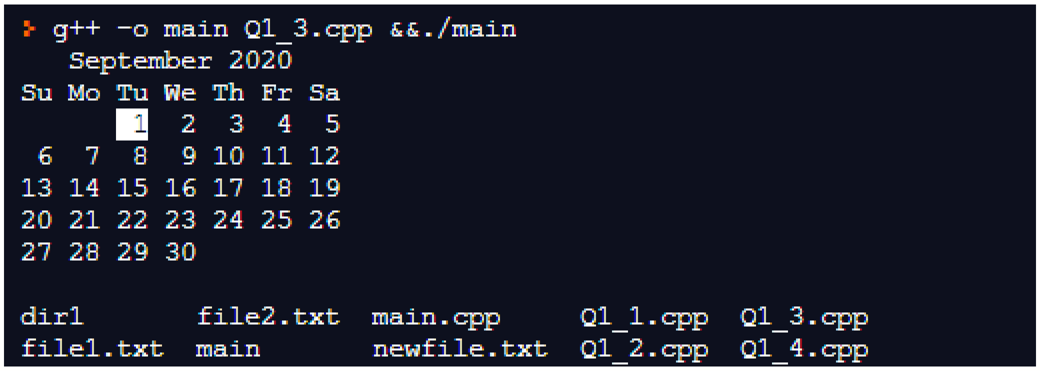
3) different programs

```
 #include<stdio.h>

#include<stdlib.h>

#include<unistd.h>
#include<sys/types.h>
//different program
int main(void)
{
    int a;
    a=fork();
    if(a<0)
    {
      fprintf(stderr,"Error in fork()\n");
      exit(-1);
    }
    else if(a>0)
    {
     execlp("/bin/ls","ls",NULL);
    }
    else
      execlp("/usr/bin/cal","cal",NULL);
    return 0;
}
```

<u>OUTPUT:</u>


```
 g++ -o main Q1_3.cpp &&./main
   September 2020
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30

dir1       file2.txt  main.cpp     Q1_1.cpp  Q1_3.cpp
file1.txt  main       newfile.txt  Q1_2.cpp  Q1_4.cpp
```

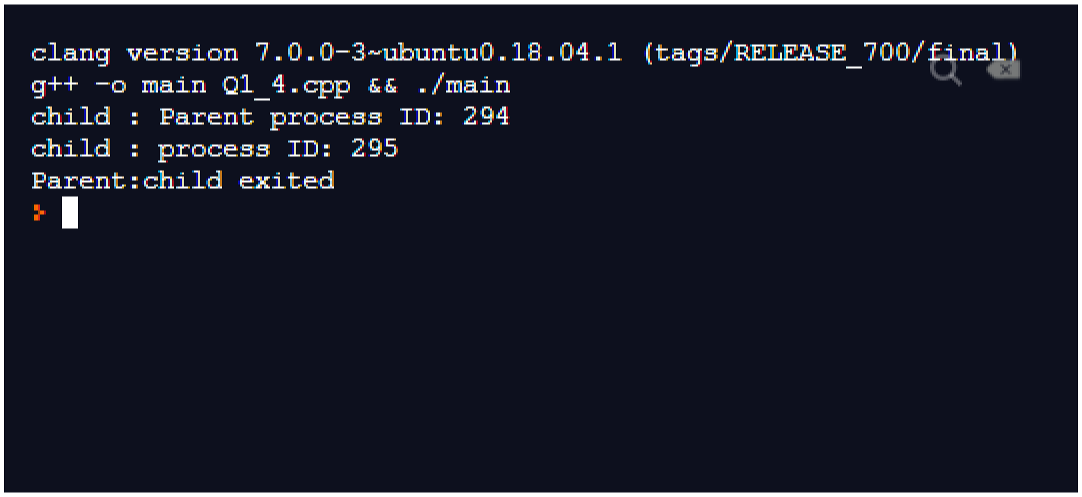
4) before terminating, the parent waits for the child to finish its task

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/types.h>
#include<sys/wait.h>
int main()
{
    int a;
    a=fork();
    if(a<0)
    {
      printf("Error in the fork");
      exit(-1);
    }
    else if(a>0){
         wait(NULL);
         printf("Parent:child exited\n");
    }

    else
    {
      printf("child : Parent process ID: %d\n",getppid());
      printf("child : process ID: %d\n",getpid());
    }

    return 0;
}
```

OUTPUT:



```
clang version 7.0.0-3~ubuntu0.18.04.1 (tags/RELEASE_700/final)
g++ -o main Q1_4.cpp && ./main
child : Parent process ID: 294
child : process ID: 295
Parent:child exited
> 
```