1. Sample program with output

#include <stdio.h>

#include<stdlib.h>

#include<unistd.h>

int num = 0;

int main(int argc, char\*argv[]){

int pid;

pid = fork();

printf("%d", num);

if(pid == 0){ /\*child\*/

num = 1;

}else if(pid > 0){ /\*parent\*/

num = 2;

}

printf("%d", num);

}

Output:

0201

**2.Program with exec/execlp**

#include<stdio.h>

#include<unistd.h>

#include<stdlib.h>

int main(){

pid\_t pid;

int i;

pid = fork(); //create a child process

if(pid == -1){

printf("Fork uncessful\n");

exit(EXIT\_FAILURE);}

else if(pid == 0){/\* child \*/

execlp("/bin/ls","ls",NULL);

for( i=5; i < 10; i++ ){

printf( "CHILD %d\n", i );

}

exit(EXIT\_SUCCESS);

}

else{/\* parent \*/

for( i=0; i < 10; i++ ){

sleep(5);

printf("\t\tPARENT %d\n", i);

}

3. Guarantees the child process will print its message before the parent process.

#include <stdio.h>

#include <sys/wait.h>

int main(void)

{

int pid;

int status;

printf("Hello World!\n");

pid = fork( );

if (pid == -1) /\* check for error in fork \*/

{

perror("bad fork"); exit(1);

}

if (pid == 0) {

printf("I am the child process.\n");

exit(1);

}

else

{

wait(&status); /\* parent waits for child to finish \*/

printf("I am the parent process.\n");

}

}

**Getpid() and getppid():**

#include <stdio.h>

#include <sys/wait.h>

#include<stdlib.h>

#include<unistd.h>

int num = 0;

int main(int argc, char\*argv[]){

int pid;

int status;

pid = fork();

printf("%d \n", num);

if(pid == 0){ /\*child\*/

num = 1;

printf("parent process id %d\n",getppid());

printf("process id %d\n", getpid());

printf("child process\n");

exit(1);

}else if(pid > 0){ /\*parent\*/

wait(&status);

num = 2;

printf("pid %d\n",pid);

printf("process id is %d\n", getpid());

printf("parent process\n");

}

printf("%d\n", num);

return 0;

}

4. **Predict the output**

#include<stdio.h>

#include<unistd.h>

#include<sys/types.h>

int main()

{

if (fork() || fork())

fork();

printf("1");

}

5. **Predict the output**

#include <stdio.h>

#include<sys/types.h>

#include<stdlib.h>

#include<unistd.h>

int i = 1;

int main() {

while(i < 4 && fork() != 0) {

i++;

}

printf("%d ", i);

exit(0);

}