# HOMEWORK 2


# HAMMAD KHAN MUSAKHEL

# 21801175


# CS342-002

# 21/02/2021

**Q1)**

```
#include<stdio.h>
#include <sys/types.h>
#include <unistd.h>

int main()
{
   int k = 5;
   pid_t  n; // stores process id

   fork();
   fork();
   fork();
   fork();
   fork();
   printf("[child] pid %d from [parent] pid %d\n",getpid(),getppid());
   for(int i; i<2^k; i++){
   wait();
   }
}
```

**Q2)**

```
void*           stack
atomic_t        usage
unsigned int flags
unsigned int ptrace
int             on_rq
unsigned int policy
int             exit_state
unsigned int jobctl
```

unsigned int personality

pid_t pid

**Q3)**

Answer: 2

**Q4)**

100

200

200

200

250

250

250

**Q5)**

```
#include<stdio.h>
#include <sys/types.h>
#include <unistd.h>


int main()
{
  pid_t  n; // stores process id

        n = fork();
        if (n < 0) {
                fprintf(stderr, "Fork Failed");
                exit(-1);
        }
        else if (n == 0) { /* child process code*/
                execlp("ls", "ls", "-al", NULL);

        }else{ /* parent process */

              pid_t x = fork();
```

```
            if( x < 0){
               fprintf(stderr, "Fork Failed");
               exit(-1);
            }
            else if(x == 0){

               execlp("ps", "ps", "aux", NULL);
             }
            else{
               wait();
               wait();
               printf ("Child Complete");
               exit(0);
            }
        }
}
```