PROGRAM 2:

(i) fork(), getpid(), wait(), exit()

```
#include<stdio.h>
#include<unistd.h>
#include<stdlib.h>

int main()
{
int x,y;
x = fork();
if(x==0){
printf("PID of child process is %d and its parent is %d\n",getpid(),getppid());
printf("In child Process X value is %d\n",x);
exit(0);
} else {
wait();
printf("PID of Parent process is %d and its parent is %d\n",getpid(),getppid());
printf("In Parent Process X value is %d\n",x);
y = fork();
if(y==0){
printf("PID of child process is %d and its parent is %d\n",getpid(),getppid());
printf("In child Process X value is %d\n",x);
exit(0);
}
else{
wait();
printf("PID of Parent process is %d and its parent is %d\n",getpid(),getppid());
printf("In Parent Process X value is %d\n",x);
exit(0);
}
}
}
```

(ii) ADD

```
#include<stdio.h>
#include<stdlib.h>
int main(int argc, char *argv[])
{
int a,b,c;
a = atoi(argv[1]);
b = atoi(argv[2]);
c= a+b;
printf("The value of argc is %d\n",argc);
printf("The value of c is %d\n",c);
exit(0);
}
```

(iii) execl

```
#include<stdio.h>
#include<unistd.h>
#include<stdlib.h>
#include<sys/wait.h>
int main(int argc, char *argv[])
{
pid_t pid;
pid = fork();
if(pid==0){
printf("Its a child Process\n");
execl("add","add",argv[1],argv[2], (char *)0);
exit(0);
}
else{
wait(0);
printf("Its a Parent Process\n");
exit(0);
}
}
```