P1.c

1 /\*The rand() function returns a pseudo-random integer in the range 0 to RAND\_MAX inclusive

2 (i.e., the mathematical range [0, RAND\_MAX]).

3 The srand() function sets its argument as the seed for a new sequence of pseudo-random integers

4 to be returned by rand(). These sequences are repeatable by calling srand() with the same seed

5 value.\*/

6

7 #include<stdio.h>

8 #include<stdlib.h>

9 main()

10 {

11 //srand(101);

12 srand(getpid());

13 int r1,r2,r3;

14 r1=rand();

15 printf("r1=%d\n",r1);

16 r2=rand();

17 printf("r2=%d\n",r2);

18 r3=rand();

19 printf("r3=%d\n",r3);

20 }

P2.c

1 //range 0 to 999

2 #include<stdio.h>

3 #include<stdlib.h>

4 main()

5 {

6 srand(getpid());

7 int a[5],i;

8 for(i=0;i<5;i++)

9 a[i]=rand()%1000;

10 printf("random numbers are...\n");

11 for(i=0;i<5;i++)

12 printf("%d\n",a[i]);

13 }

P3.c

1 //range 0 to 9

2 //range 1 to 10

3 #include<stdio.h>

4 #include<stdlib.h>

5 main()

6 {

7 srand(getpid());

8 int a[5],i;

9 for(i=0;i<5;i++)

10 a[i]=rand()%10+1; //(1 t0 10)

11 printf("random numbers are...\n");

12 for(i=0;i<5;i++)

13 printf("%d\n",a[i]);

14 }

P4.c

1 #include<stdio.h>

2 #include<stdlib.h>

3 #include<unistd.h>

4 main()

5 {

6 if(fork()==0)

7 {

8 int t1;

9 srand(getpid());

10 t1=rand()%10+1; //1 to 10

11 printf("child process goes to delay of %d sec\n",t1);

12 sleep(t1);

13 printf("child process terminated...\n");

14 }

15 else

16 {

17 int t2;

18 srand(getpid());

19 t2=rand()%10+1; //1 to 10

20 printf("parent process goes to delay of %d sec\n",t2);

21 sleep(t2);

22 printf("prent process terminated...\n");

23 }

24 }