PROGRAM for PROBLEM 3 and PROBLEM 4:

```
#include <stdio.h>
#include <limits.h>
#include <math.h>

int findRange(double r, int *range) {
	int i = 0;
	for (i = 1; i < 11; i++) {
		double ra = ((double)i)/((double)10);
		if (r < ra) return i-1;
	}
	return -1;
}

int main() {
	double m = 2147483647;
	double k = 16807;
	double s = 1111;
	double r = 0;
	double umean = 0;
	double uvar = 0;
	double expMean = 0;
	double expVar = 0;

	int range[10] = {0,};

	int i = 0;
	for (i = 0; i < 10000; i++) {
		s = fmod(k*s, m);
		r = s/m;
		umean+=r;
		uvar+=r*r;
		double expY = (-0.5)*(log(r));
		expMean+=expY;
		expVar+=expY*expY;
		int j = findRange(r, range);
		if (j >= 0) {
			range[j] += 1;
		}
	}

	printf("Mean(Uniform): %lf\n", umean/10000);
	printf("Variance(Uniform): %lf\n", (uvar/10000 - ((umean/10000)*(umean/10000))));

	printf("Mean(Exponential): %lf\n", expMean/10000);
	printf("Variance(Exponential): %lf\n", (expVar/10000 - ((expMean/10000)*(expMean/10000))));
	for (i = 0; i < 10; i++) {
		float range1 = (float)i/10;
		float range2 = ((float)i+1)/10;
		printf("Range[%f, %f]: %d\n", range1, range2, range[i]);
```

```
    }

    return 0;
}
```

PROGRAM for PROBLEM 5:

```
#include <stdio.h>
#include <math.h>
#include <limits.h>

double getXRandom(double m, double k, double *s) {
    (*s) = fmod(k*(*s), m);
    double r = (*s)/m;
    return r;
}

int main() {
    double m = 2147483647;
    double k = 16807;
    double s = 1111;
    double r1 = 0;
    int i = 0;
    int rzero = 0;
    int rone = 0;
    int szero = 0;
    int sone = 0;
    int zeroincorrect = 0;
    int zerocorrect = 0;
    int oneincorrect = 0;
    int onecorrect = 0;
    int error = 0;
    for (i = 0; i < 10000; i++) {
        int bit = 0;
        r1 = getXRandom(m,k,&s);
        if (r1 < 0.45) {
            bit = 0;
        } else {
            bit = 1;
        }

        double r2 = getXRandom(m,k,&s);
        int rbit = 0;
        if (bit == 0) {
            if (r2 < 0.15) {
                rbit = 1;
                oneincorrect+=1;
            } else {
                rbit = 0;
                zerocorrect += 1;
```

```
            }
        } else {
            if (r2 < 0.05) {
                rbit = 0;
                zeroincorrect += 1;
            } else {
                rbit = 1;
                onecorrect += 1;
            }
        }
        szero += (bit == 0)?1:0;
        sone += (bit == 1)?1:0;
        rzero += (rbit == 0)?1:0;
        rone  += (rbit == 1)?1:0;
        error += (bit == rbit)?0:1;
    }
    printf("RZ: %d RO: %d SZ: %d SO: %d error: %d\n", rzero, rone, szero, sone, error);
    printf("RZEROCORRECT: %d RZEROINCORRECT: %d RONECORRECT: %d RONEINCORRECT: %d\n",
zerocorrect, zeroincorrect, onecorrect, oneincorrect);
    printf("P(0 received): %lf\n", (double)rzero/(double)10000);
    printf("P(1 was transmitted| 1 was received): %lf\n", (double)onecorrect/(double)rone);
    printf("P(error): %lf\n", (double)error/(double)10000);
    return 1;
}
```