answers09

1.

```
#include <stdio.h>
bool isRelativePrime(int a, int i);
int main()
{
    int count = 0;
    for (int a=2; a<=100; a++) {
        for (int b=a+1; b<=100; b++) {
            if ( !isRelativePrime(b, a) )
                continue;
            for (int c=b+1; c<=100; c++) {
                if (isRelativePrime(c, a) && isRelativePrime(c, b)) {
                    printf("%d %d %d\n", a, b, c);
                    count++;
                }
            }
        }
    }
    printf("%d\n", count);
}

bool isRelativePrime(int m, int n) {
    while ( m % n != 0 ) {
        m = m % n;
        int tmp = m;
        m = n;
        n = tmp;
    }
    return n==1;
}
```

2.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
double evalProbBySim(int N, int k, int T);
int main()
{
    int N, T = 1000000;
    srand((unsigned int)time(NULL));
    for (int k = 1; k<=6; k++) {
        printf("%d: %lf\n", k, evalProbBySim(6*k, k, T));
    }
}

double evalProbBySim(int N, int k, int T)
{
    int count = 0;
    for (int i=0; i<T; i++) {
        int hit = 0;
        for (int t=0; t<N; t++) {
            int rd = rand() % 6 + 1;
            if (rd == 1)
                hit++;
        }
        if (hit >= k)
            count++;
    }
    return  (double)count/T;
}
```

3.

```
#include <stdio.h>
int lenOfOverlap( int sa, int ta, int sb, int tb );
int main()
{
    int x1, y1, x2, y2, x3, y3, x4, y4;
    scanf("%d %d %d %d %d %d %d %d", &x1, &y1, &x2, &y2, &x3, &y3, &x4, &y4);
    int sx1, tx1, sy1, ty1, sx2, tx2, sy2, ty2;

    if (x1 > x2) { sx1 = x2; tx1 = x1; }
    else { sx1 = x1; tx1 = x2; }

    if (y1 > y2) { sy1 = y2; ty1 = y1; }
    else { sy1 = y1; ty1 = y2; }

    if (x3 > x4) { sx2 = x4; tx2 = x3; }
    else { sx2 = x3; tx2 = x4; }

    if (y3 > y4) { sy2 = y4; ty2 = y3; }
    else { sy2 = y3; ty2 = y4; }

    int width = lenOfOverlap(sx1, tx1, sx2, tx2);
    int height = lenOfOverlap(sy1, ty1, sy2, ty2);

    printf("%d\n", width*height);
}

int lenOfOverlap( int sa, int ta, int sb, int tb )
{
    int s, t;
    if (sa >= sb) s = sa;
    else s = sb;

    if (ta >= tb) t = tb;
    else t = ta;

    if ( s > t ) return 0;
    else return t-s;
}
```

4.

```
#include <stdio.h>
#define MAX 100
int compareInts( int sa, int ta, int sb, int tb);
int main()
{
    int n = 0, start[MAX], end[MAX];
    FILE *fp = fopen("input4.txt", "r");
    while (!feof(fp)) {
        fscanf(fp, "%d", &start[n]);
        fscanf(fp, "%d", &end[n++]);
    }
    fclose(fp);

    for (int i=n-1; i>0; i--) {
        for (int j=0; j<i; j++) {
            if ( compareInts(start[j], end[j], start[j+1], end[j+1]) > 0 ) {
```

```c
                int tmp = start[j];
                start[j] = start[j+1];
                start[j+1] = tmp;
                tmp = end[j];
                end[j] = end[j+1];
                end[j+1] = tmp;
            }
        }
    }

    for (int i=0; i<n; i++)
        printf("%d %d\n", start[i], end[i]);
}

int compareInts( int sa, int ta, int sb, int tb )
{
    if (sa > sb || sa == sb && ta > tb) return 1;
    else if ( sa == sb && ta && tb ) return 0;
    else return -1;
}
```

5.

```c
#include <stdio.h>
#define MAX 100
int  compareLexicographic(int p, int q)
{
    int digits1[20], digits2[20];
    int k = 0;    // number of digits of p
    while(p > 0) {
        digits1[k++] = p%10;
        p /= 10;
    }

    int h = 0;   // h is # of digits of q
    while(q > 0) {
        digits2[h++] = q%10;
        q /= 10;
    }

    for (int x = k-1, y = h-1; x >= 0 && y >= 0; x--, y--) {
        if (digits1[x] < digits2[y]) {
            return -1;
        }
        else if (digits1[x] > digits2[y]) {
            return 1;
        }
    }
    if ( k > h ) return 1;
    else if ( k < h ) return -1;
    else return 0;
}

int main()
{
    int data[MAX], n=0;
    FILE *fp = fopen("input5.txt", "r");
    while (!feof(fp)) fscanf(fp, "%d", &data[n++]);
    fclose(fp);

    /* insertion sort */
    for (int i=1; i<n; i++) {
        int j = i-1, tmp = data[i];
        while( j >= 0 && compareLexicographic(tmp, data[j]) < 0) {
            data[j+1] = data[j];
            j--;
        }
        data[j+1] = tmp;
    }

    for (int i=0; i<n; i++)
        printf("%d ", data[i]);
}
```

7.(Not programming assignment)

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define MAX_NUMBER 100
void initialize_number_generator(void);
int play(int secret_number);
int choose_secret_number(void);
int main(void)
{
    char command;
    int secret_number, best_record = 100;
    initialize_number_generator();
    printf("Guess the secret number between 1 and %d.\n\n", MAX_NUMBER);
    while(1)
    {
        secret_number = choose_secret_number();
        int record = play(secret_number);
        if (record < best_record) {
            best_record = record;
            printf("Congratulation ! You broke the world record.\n");
        }
        printf("Play again? (Y/N) ");
        scanf(" %c", &command);
        if (command != 'y' && command != 'Y')
            break;
        printf("\n");
    }
    return 0;
}

int choose_secret_number(void)
{
    int secret_number;
    secret_number = rand() % MAX_NUMBER + 1;
    return secret_number;
}

int play(int secret_number)
{
    int guess, num_guesses = 0;

    while(1) {
        printf("Enter guess: ");
        scanf("%d", &guess);
        num_guesses++;
        if (guess == secret_number) {
            printf("You won in %d guesses!\n\n", num_guesses);
            return num_guesses;
        }
        else if (guess < secret_number)
            printf("Too low; try again.\n");
        else
            printf("Too high; try again.\n");
    }
}

void initialize_number_generator(void)
{
    srand((unsigned int) time(NULL));
}
```