

staircase.c

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

```
int row_staircase();
```

```
void last_row(int n);
```

```
void main(){
```

```
    int n, i;
```

```
    int remainder = n%5;
```

```
    int rows;
```

```
    printf("Enter number of staircases (must be in [0,100]): ");
```

```
    scanf(" %d", &n);
```

```
    rows = n/5;
```

```
    if ((n>=0) && (n<=100)){
```

```
        for (i=0; i<rows; i++){
```

```
            row_staircase();
```

```
        }
```

```
        if (remainder !=0){
```

```
            last_row(n);
```

```
        }
```

```
        else return;
```

```
    }
```

```
    else{
```

```
        printf("n=%d, must be in [0,100]\n",n);
```

```
    }
```

```
    return;
```

```
}
```

```
int row_staircase(){
```

```
    printf(" # # # # #\n");
```

```
    printf(" ## ## ## ## ##\n");
```

```
    printf("### ### ### ### ###\n");
```

```
    return 0;
```

```
}
```

```
void last_row(int n){
```

```
    int i, remainder;
```

```
    remainder = n%5;
```

```
    for (i=0; i<remainder; i++){
```

```
        printf(" # ");
```

```
    }
```

```

printf("\n");

for (i=0; i<remainder; i++){
printf(" ## ");
}
printf("\n");

for (i=0; i<remainder; i++){
printf("### ");
}
printf("\n");
}

```

staircase2.c

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

```

void create_lines(int i, int m);
void row_staircase(int p, int m);

```

```

void main(){
    unsigned int n, m, p;
    int i;
    int remainder = n%5;
    int rows;

    printf("Enter number of: staircases, stairs, rows: ");
    scanf(" %d %d %d", &n, &m, &p);

    rows = n/p;
    remainder = n%p;
    if (((n>=0) && (n<=100)) && ((m>=1) && (m<=9)) && ((p>=1) && (p<=100))){
        for (i=0; i<rows; i++){
            row_staircase(p, m);
        }
        if (remainder != 0){
            row_staircase(remainder, m);
        }
        else return;
    }
    if (((n<0) || (n>100)) && ((m>=1) && (m<=9)) && ((p>=1) && (p<=100))){
        printf("n=%d, must be in [0,100]\n", n);
    }
}

```

```

    if (((n>=0) && (n<=100)) && ((m<1) || (m>9)) && ((p>=1) && (p<=100))) {
        printf("m=%d, must be in [1,9]\n", m);
    }
    if (((n>=0) && (n<=100)) && ((m>=1) && (m<=9)) && ((p<1) || (p>100))) {
        printf("p=%d, must be in [1,100]\n", p);
    }
    if (((n<0) || (n>100)) && ((m<1) || (m>9)) && ((p>=1) && (p<=100))) {
        printf("n=%d, must be in [0,100]\n", n);
        printf("m=%d, must be in [1,9]\n", m);
    }
    if (((n<0) || (n>100)) && ((m>=1) && (m<=9)) && ((p<1) || (p>100))) {
        printf("n=%d, must be in [0,100]\n", n);
        printf("p=%d, must be in [1,100]\n", p);
    }
    if (((n>=0) && (n<=100)) && ((m<1) || (m>9)) && ((p<1) || (p>100))) {
        printf("m=%d, must be in [1,9]\n", m);
        printf("p=%d, must be in [1,100]\n", p);
    }
    if (((n<0) || (n>100)) && ((m<1) || (m>9)) && ((p<1) || (p>100))) {
        printf("n=%d, must be in [0,100]\n", n);
        printf("m=%d, must be in [1,9]\n", m);
        printf("p=%d, must be in [1,100]\n", p);
    }
    return;
}

```

```

void create_lines(int i, int m) { //print i th line for staircase with m stairs
    int j;
    switch (i) {
        case 1:
            for (j=0; j<(m-1); j++) {
                printf(" ");
            }
            printf("%d", m);
            printf(" ");
            break;

        case 2:
            for (j=0; j<(m-2); j++) {
                printf(" ");
            }
            for (j=0; j<2; j++) {
                printf("%d", m);
            }

```

```

}
printf(" ");
break;

case 3:
for (j=0; j<(m-3);j++){
    printf(" ");
}
for (j=0;j<3;j++){
    printf("%d",m);
}
printf(" ");
break;

case 4:
for (j=0; j<(m-4);j++){
    printf(" ");
}
for (j=0;j<4;j++){
    printf("%d",m);
}
printf(" ");
break;

case 5:
for (j=0; j<(m-5);j++){
    printf(" ");
}
for (j=0;j<5;j++){
    printf("%d",m);
}
printf(" ");
break;

case 6:
for (j=0; j<(m-6);j++){
    printf(" ");
}
for (j=0;j<6;j++){
    printf("%d",m);
}
printf(" ");
break;

```

```

case 7:
for (j=0; j<(m-7);j++){
    printf(" ");
}
for (j=0;j<7;j++){
    printf("%d",m);
}
printf(" ");
break;

```

```

case 8:
for (j=0; j<(m-8);j++){
    printf(" ");
}
for (j=0;j<8;j++){
    printf("%d",m);
}
printf(" ");
break;

```

```

case 9:
for (j=0; j<(m-9);j++){
    printf(" ");
}
for (j=0;j<9;j++){
    printf("%d",m);
}
printf(" ");
break;
}
}

```

```

void row_staircase(int p, int m){
    int i,j;
    for (i=1;i<(m+1);i++){
        for (j=0;j<p;j++){
            create_lines(i,m);
        }
        printf("\n");
    }
    return;
}

```

factors.c

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

```
int main(){
    unsigned long x,y;
    unsigned long i,n;
    unsigned long sum = 0;
    unsigned long GCF=1;
    unsigned long LCM;

    printf("Enter two positive integers: ");
    scanf(" %lu %lu", &x, &y);

    if (x>y){
        n = y;
    }
    else{
        n = x;
    }
    printf("Common factors of (%lu,%lu): ",x,y);
    for (i=1;i<=n;i++){
        if ((x%i==0) && (y%i==0)){
            printf("%lu ",i);
            sum += i;
            if (GCF<i){
                GCF = i;
            }
        }
    }

    printf("\n");

    LCM = ((x * y) / GCF);

    sum += (LCM + GCF);

    printf("The greatest common factor(GCF) of (%lu,%lu): %lu\n",x,y,GCF);
    printf("The least common multiple(LCM) of (%lu,%lu): %lu\n",x,y,LCM);
    printf("The sum of all common factors, the GCF, the LCM: %lu\n",sum);

    return 0;
}
```

reverse_long.c

#include <stdio.h>

```
int main(){
    unsigned long int x;
    long int new_num = 0;
    printf("Enter a integer: ");
    scanf(" %ld", &x);
    while (x!=0){
        new_num = new_num * 10;
        new_num += x%10;
        x = x/10;
    }
    printf("%ld\n",new_num);
    return 0;
}
```

reverse_dec_string.c

#include <stdio.h>

#include <string.h>

```
int main(){
    char x[20];
    int i;
    int lead, trail;

    printf("Enter decimal string: ");
    scanf("%s", x);

    if ((strlen(x)-1)>=20){
        trail = 19;
    }
    else{
        trail = strlen(x)-1;
    }

    lead = 0;

    while (x[lead]!='0'){
        lead++;
    }

    while (x[trail]!='0'){
```

```
    trail--;  
    }  
  
    for (i=trail;i>=lead;i--){  
        printf("%c",x[i]);  
    }  
    printf("\n");  
  
    return 0;  
}
```