Week\_1

1.

#include <stdio.h>

int greater\_common\_divsor(int,int);

int least\_common\_multiple(int,int);

int main(void) {

int a,b;

puts("Input two num(No check.):");

scanf("%d%d",&a,&b);

printf("greater\_common\_divsor:%d\n",greater\_common\_divsor(a,b));

printf("least\_common\_multiple:%d\n",least\_common\_multiple(a,b));

return 0;

}

int greater\_common\_divsor(int a,int b){

if(a < b){//ensure a>=b

a ^= b;

b ^= a;

a ^= b;

}

int r;

while (r = a % b) {

a = b;

b = r;

}

return b;

}

int least\_common\_multiple(int a,int b){

return (a \* b / greater\_common\_divsor(a , b));

}

2.

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

double average(const double \*,int);

double variance(const double \*,int);

int main(void) {

int n;

double \* arr\_pointer;

puts("Input N:");

scanf("%d",&n);

arr\_pointer = (double \*)malloc(n\*sizeof(double));

puts("Input sample:");

for(int i = 0;i < n;i ++)

scanf("%lf",&arr\_pointer[i]);

printf("Average:%lf Variance:%lf\n",average(arr\_pointer,n),

variance(arr\_pointer,n));

return 0;

}

double average(const double \* arr,int length){

double sum = 0;

for(int i = 0;i < length;i ++)

sum += arr[i];

return (sum / length);

}

double variance(const double \* arr,int length){

double average\_value = average(arr,length);

double sum\_var = 0;

for(int i = 0;i < length;i ++)

sum\_var += pow(arr[i]-average\_value,2);

return (sum\_var / length);

}

3.

#include <stdio.h>

int main(void){

for(int i = 1;i <= 4;i ++)

for(int j = 1;j <= 4;j ++)

for(int k = 1;k <= 4;k ++)

if(i != j && i != k && j != k)

printf("%d\n",i \* 100 + j \*10 + k);

return 0;

}

4.

#include <stdio.h>

#define TREE\_ARGS 9

#define TREE\_FLOOR 3

void show\_space(int);

void show\_star(int);

void show\_tree(void);

void show\_trangle(int,int);

void show\_rect(int,int);

int main(void) {

//ugly tree emmmm.

show\_tree();

return 0;

}

void show\_space(int x){

for(int i = 0;i < x;i ++)

putchar(' ');

}

void show\_star(int x){

for(int i = 0;i < x;i ++)

putchar('\*');

}

void show\_trangle(int x,int num){

for(int i = x / 2;i >= num;i --){

show\_space(i);

show\_star(x - i \* 2);

putchar('\n');

}

}

void show\_tree(void){//x is odd

int x = TREE\_ARGS;

int num = TREE\_FLOOR;

while(num -- > 0)

show\_trangle(x,num);

show\_rect(x,x/2+1);

}

void show\_rect(int x,int y){

for(int i = 0;i < y; i++){

show\_space(x / 2 - 1);

show\_star(3);

putchar('\n');

}

show\_star(x);

putchar('\n');

show\_star(x);

putchar('\n');

}

5.

#include <stdio.h>

int n = 0;

void swap(int \*, int \*);

void perm(int \*, int, int);

int main(void){

int list[] = {1, 2, 3, 4, 5, 6, 7, 8, 9};

perm(list, 0, 8);

printf("total:%d\n", n);

return 0;

}

void swap(int \*a, int \*b){

int m;

m = \*a;

\*a = \*b;

\*b = m;

}

void perm(int list[], int k, int m){

if(k > m){

if(list[0] \* 100 + list[1] \* 10 + list[2] \* 1 +

list[3] \* 100 + list[4] \* 10 + list[5] \* 1 ==

list[6] \* 100 + list[7] \* 10 + list[8] \* 1){

printf("%d%d%d + %d%d%d = %d%d%d", list[0],list[1],list[2],list[3],list[4],list[5],list[6],list[7],list[8]);

printf("\n");

n++;

}

}else{

for(int i = k; i <= m; i++){

swap(&list[k], &list[i]);

perm(list, k + 1, m);

swap(&list[k], &list[i]);

}

}

}

6.

#include <stdio.h>

int getSeedNumber(int n){

int num=0;

if(n==10){

printf("Day:%d Leaf:%d\n", n, 1);

return 1;

}else{

num = (getSeedNumber(n+1)+1)\*2 ; //Problem

printf("Day:%d Leaf:%d\n", n, num);

}

return num;

}

int main(){

int num = getSeedNumber(1);

printf("All :%d \n", num);

return 0;

}