\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Report: HW5\_1

Author: C24089036 楊佳恩 <[charles870126@gmail.com](mailto:charles870126@gmail.com)>

Class: 物理系

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Code:

#include <stdio.h>

#include <stdlib.h>

int purely\_recursive(int n);

int iterative(int n);

int modifited\_recursive(int n,int\* a);

int main()

{

int n = 5;

printf("%d\n", purely\_recursive(n));

printf("%d\n", iterative(n));

int a[n+1], i;

for(i=0; i<n+1; i++) a[i]=0;

printf("%d\n", modifited\_recursive(n,a));

return 0;

}

int purely\_recursive(int n)

{

if (n <= 2) return n;

return purely\_recursive(n-1) + purely\_recursive(n-2) + purely\_recursive(n-3);

}

int iterative(int n)

{

int f0 = 0, f1 = 1, f2 = 2, fn, i;

for (i = 2; i < n; i++) {

fn = f0 + f1 + f2;

f0 = f1;

f1 = f2;

f2 = fn;

}

return fn;

}

Compilation:

gcc hw5\_1.c -o hw5\_1

Execution:

./hw5\_1

Output:

11

11

11

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Report: HW5\_2

Author: C24089036 楊佳恩 <[charles870126@gmail.com](mailto:charles870126@gmail.com)>

Class: 物理系

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Code:

#include<stdio.h>

void cover(int a[][2], int r, int g[][2]);

void Union(int a[][2]);

int main()

{

int v = 18;

int r[4][2];

int I[4][2] = { 3, 19,

11, 33,

18, 80,

80, 100};

cover(I, v, r);

Union(I);

return 0;

}

void cover(int I[][2], int v, int r[][2])

{

printf("R = ");

int i;

for (i = 0; i < 4; i++)

{

if (v >= I[i][0] && v <= I[i][1])

{

r[i][0] = I[i][0];

r[i][1] = I[i][1];

printf("[%d, %d] ", r[i][0], r[i][1]);

}

}

printf("\n");

}

void Union(int I[][2])

{

int i, j;

int t[4][2];

for (i = 0; i < 4; i++)

{

t[i][0] = I[i][0] - 1;

t[i][1] = I[i][1];

}

int T[8], k = 0;

for (i = 0; i < 4; i++)

{

for (j = 0; j < 2; j++)

{

T[k] = t[i][j];

k++;

}

}

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

{

for (j = 0; j < 8; j++)

{

if (T[j] > T[i])

{

k = T[j];

T[j] = T[i];

T[i] = k;

}

}

}

printf("T = ");

for (i = 0; i < 8; i++) printf("%d ", T[i]);

printf("\n");

}

Compilation:

gcc hw5\_2.c -o hw5\_2

Execution:

./hw5\_2

Output:

R = [3, 19] [11, 33] [18, 80]

T = 2 10 17 19 33 79 80 100

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Report: HW5\_3

Author: C24089036 楊佳恩 <[charles870126@gmail.com](mailto:charles870126@gmail.com)>

Class: 物理系

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Code:

#include<stdio.h>

#define TREE\_SIZE 31

int a[TREE\_SIZE + 1] = { 15, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 };

int tree\_size = 15;

void interative(int\* a, int c);

void mirror\_recursive(int root\_index, int a[], int n);

unsigned compute\_tree\_height(unsigned size);

int main()

{

mirror\_recursive(8, a, a[0]);

interative(a, 4);

int i;

for(i = 0; i < 16; i++) printf("%d ", a[i]);

return 0;

}

void interative(int\* a, int c)

{

int g = 0, b, i;

for (b = 0; b < c - 1; b++)

{

for (i = 0; i < (1 << b); i++)

{

g = a[(1 << b + 1) + i];

a[(1 << b + 1) + i] = a[(1 << b + 1) + (1 << b + 1) - 1 - i];

a[(1 << b + 1) + (1 << b + 1) - 1 - i] = g;

}

}

}

void mirror\_recursive(int root\_index, int a[], int n)

{

int i, j, k;

int b[TREE\_SIZE];

if (root\_index \* 2>a[0])return;

for (i = 1; i<8; i++){

if (root\_index\*(1 << i)>a[0]) break;

for (j = 0; j<(1 << i) / 2; j++){

b[j] = a[(1 << i)\*root\_index + j];

a[(1 << i)\*root\_index + j] = a[(1 << i)\*root\_index + (1 << i) / 2 + j];

a[(1 << i)\*root\_index + (1 << i) / 2 + j] = b[j];

}

}

mirror\_recursive(2 \* root\_index, a, a[0]);

mirror\_recursive(2 \* root\_index + 1, a, a[0]);

}

Compilation:

gcc hw5\_3.c -o hw5\_3

Execution:

./hw5\_3

Output:

15 1 3 2 7 6 5 4 15 14 13 12 11 10 9 8

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Report: HW5\_4

Author: C24089036 楊佳恩 <[charles870126@gmail.com](mailto:charles870126@gmail.com)>

Class: 物理系

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Code:

#include<stdio.h>

void ip\_to\_pattern(int ip[4]);

void pattern\_to\_ip(int pattern[4][8]);

int main()

{

int ip[4] = {129, 160, 96, 1};

int pattern[4][8] = {{ 1, 0, 0, 0, 0, 0, 0, 1 },

{ 1, 0, 1, 0, 0, 0, 0, 0 },

{ 0, 1, 1, 0, 0, 0, 0, 0 },

{ 0, 0, 0, 0, 0, 0, 0, 1 }};

ip\_to\_pattern(ip);

pattern\_to\_ip(pattern);

return 0;

}

void ip\_to\_pattern(int ip[])

{

int a[4][8] = {0};

int i, j;

for (j = 0; j < 4; j++)

{

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

{

if (ip[j] & 1 << i)

{

a[j][i] = 1;

}

}

}

for (j = 0; j < 4; j++)

{

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

{

printf("%d", a[j][7 - i]);

}

printf(" ");

}

printf("\n");

}

void pattern\_to\_ip(int pattern[4][8])

{

int i, j, k;

int a[4] = {0};

for (i = 0; i < 4; i++)

{

for (j=7,k=0 ; j > -1 ; k++,j--)

{

if (pattern[i][j] == 1)

{

a[i] = a[i] | (1 << k);

}

}

printf("%d.",a[i]);

}

}

Compilation:

gcc hw5\_4.c -o hw5\_4

Execution:

./hw5\_4

Output:

10000001 10100000 01100000 00000001

129.160.96.1.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Report: HW5\_5

Author: C24089036 楊佳恩 <[charles870126@gmail.com](mailto:charles870126@gmail.com)>

Class: 物理系

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Code:

#include <stdio.h>

#define N 4

int boolfunc(int \*var, int m);

int recursivebool(int \*var, int n);

main()

{

int varbool[20];

recursivebool(varbool, N);

}

int boolfunc(int \*var, int m)

{

int result = var[0], i;

for (i = 1; i<m; i++) result = (result && var[i]);

return result;

}

int recursivebool(int \*var, int n)

{

int localvar[20], i, j;

if (n == 0){

for (i = 0; i<N; i++) printf("%d ", var[i]);

printf("%d\n", boolfunc(var, N));

return;

}

for (j = 0; j <= 1; j++) {

var[n - 1] = j;

recursivebool(var, n - 1);

}

}

Compilation:

gcc hw5\_5.c -o hw5\_5

Execution:

./hw5\_5

Output:

0 0 0 0 0

1 0 0 0 0

0 1 0 0 0

1 1 0 0 0

0 0 1 0 0

1 0 1 0 0

0 1 1 0 0

1 1 1 0 0

0 0 0 1 0

1 0 0 1 0

0 1 0 1 0

1 1 0 1 0

0 0 1 1 0

1 0 1 1 0

0 1 1 1 0

1 1 1 1 1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Report: HW5\_6

Author: C24089036 楊佳恩 <[charles870126@gmail.com](mailto:charles870126@gmail.com)>

Class: 物理系

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Code:

#include<stdio.h>

int determinant(int f[][10], int x);

int main()

{

int a[10][10] = {0};

printf("%d\n", determinant(a, 10));

return 0;

}

int determinant(int f[][10], int x)

{

int pr = 1, c[10], d = 0, b[10][10], j, p, q, t;

if (x == 2) return (f[1][1] \* f[2][2] - f[1][2] \* f[2][1]);

for (j = 1; j <= x; j++){

int r = 1, s = 1;

for (p = 1; p <= x; p++) {

for (q = 1; q <= x; q++) {

if (p != 1 && q != j) {

b[r][s] = f[p][q];

s++;

if (s > x - 1) { r++; s = 1; }

}

}

}

pr = j%2?-1:1; // 🡨look at here

c[j] = pr\*determinant(b, x - 1);

}

for (j = 1, d = 0; j <= x; j++) d += (f[1][j] \* c[j]);

return(d);

}

Compilation:

gcc hw5\_6.c -o hw5\_6

Execution:

./hw5\_6

Output:

0