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

Report: HW5\_1

Author: E94074029 江羿賢 <e94074029@gs.ncku.edu.tw>

Class: 資訊111級乙班

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

#include <stdio.h>

#include <stdlib.h>

int purely(int);

int iterative(int);

int modified(int);

int main()

{

int input = 5;

int input1 = 10;

printf("1.\n");

//test data 1

printf("test data 1\n");

printf("(1) purely recursive f(5):\n");

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

printf("(2) iterative f(5):\n");

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

printf("(3) a modified recursive function f(5):\n");

printf("%d\n", modified(input));

//test data 2

printf("\n");

printf("test data 2\n");

printf("(1) purely recursive f(10):\n");

printf("%d\n", purely(input1));

printf("(2) iterative f(10):\n");

printf("%d\n", iterative(input1));

printf("(3) a modified recursive function f(10):\n");

printf("%d\n", modified(input1));

}

int purely(int input)

{

if (input < 3)

return input;

return purely(input - 1) + purely(input - 2) + purely(input - 3);

}

int iterative(int input)

{

int a[50];

for (int input = 0; input < 3; input++)

a[input] = input;

if (input < 3)

return input;

for (int x = 3; x <= input; x++)

a[x] = a[x - 1] + a[x - 2] + a[x - 3];

return a[input];

}

int modified(int input)

{

static int y = 2;

static int a[50];

for (int i = 0; i < 3; i++)

a[i] = i;

if (input <= y)

return a[input];

else

{

a[input] = modified(input - 1) + modified(input - 2) + modified(input - 3);

y++;

return a[input];

}

}

Compilation:

gcc hw5\_1.c -o hw5\_1

Execution:

./hw5\_1

Output:

1.

test data 1

(1) purely recursive f(5):

11

(2) iterative f(5):

11

(3) a modified recursive function f(5):

11

test data 2

(1) purely recursive f(10):

230

(2) iterative f(10):

230

(3) a modified recursive function f(10):

230

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

Report: HW5\_2

Author: E94074029 江羿賢 <e94074029@gs.ncku.edu.tw>

Class: 資訊111級乙班

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

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

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

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

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

int main()

{

printf("2.\n");

//test data 1

int I[10] = {3, 19, 11, 33, 18, 80, 80, 100};

int R[10];

int T[10];

int v = 18;

memset(R, 0, 10);

memset(T, 0, 10);

printf("test data 1 I={3, 19, 11, 33, 18, 80, 80, 100},v=18\n");

printf("(a) set of ranges(R) is\n");

setR(I, v, R);

printf("\n");

printf("(b) set of union(T) is\n");

setT(I, T);

printf("\n");

printf("\n");

//test data 2

int I1[10] = {4, 20, 13, 22, 17, 52, 77, 99};

int R1[10];

int T1[10];

int v1 = 15;

memset(R1, 0, 10);

memset(T1, 0, 10);

printf("test data 2 I={4, 20, 13, 22, 17, 52, 77, 99},v=15\n");

printf("(a) set of ranges(R) is\n");

setR(I1, v1, R1);

printf("\n");

printf("(b) set of union(T) is\n");

setT(I1, T1);

printf("\n");

}

void setR(int \*I, int v, int \*R)

{

int size = 1;

int j = 0;

for (size = 1; I[size] != 0; size++)

NULL;

for (int i = 0; i < size; i += 2)

{

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

{

R[j] = I[i];

R[j + 1] = I[i + 1];

j += 2;

}

}

printf("{");

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

{

if (i != j - 1)

printf("%d,", R[i]);

else

printf("%d", R[i]);

}

printf("}");

}

void setT(int \*I, int \*T)

{

int size = 1;

for (size = 1; I[size] != 0; size++)

NULL;

for (int i = 0; i < size; i += 2)

{

T[i] = I[i] - 1;

T[i + 1] = I[i + 1];

}

for (int i = size - 1; i > 0; i--)

{

for (int j = 0; j <= i - 1; j++)

{

if (T[j] > T[j + 1])

{

swap(&T[j], &T[j + 1]);

}

}

}

printf("{");

for (int k = 0; k < size; k++)

{

if (k != size - 1)

printf("%d,", T[k]);

else

printf("%d", T[k]);

}

printf("}");

}

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

{

int temp = 0;

temp = \*a;

\*a = \*b;

\*b = temp;

}

Compilation:

gcc hw5\_2.c -o hw5\_2

Execution:

./hw5\_2

Output:

2.

test data 1 I={3, 19, 11, 33, 18, 80, 80, 100},v=18

(a) set of ranges(R) is

{3,19,11,33,18,80}

(b) set of union(T) is

{2,10,17,19,33,79,80,100}

test data 2 I={4, 20, 13, 22, 17, 52, 77, 99},v=15

(a) set of ranges(R) is

{4,20,13,22}

(b) set of union(T) is

{3,12,16,20,22,52,76,99}

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

Report: HW5\_3

Author: E94074029 江羿賢 <e94074029@gs.ncku.edu.tw>

Class: 資訊111級乙班

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

#include <stdio.h>

#include <stdlib.h>

void recursive(int \*, int);

void iterative(int \*);

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

int power(int, int);

void print(int \*, int);

int main()

{

int iter[16] = {15, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15};

int recur[16] = {15, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15};

int iter1[32] = {31, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31};

int recur1[32] = {31, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31};

printf("3.\n");

//test data 1

printf("test data 1\n");

printf("(a) Write a recursive function to mirror\n");

printf("Before mirror:\n");

print(recur, recur[0]);

recursive(recur, 1);

printf("After mirror:\n");

print(recur, recur[0]);

printf("(b) Write a iterative function to mirror\n");

printf("Before mirror:\n");

print(iter, iter[0]);

iterative(iter);

printf("After mirror:\n");

print(iter, iter[0]);

//test data 2

printf("\n");

printf("test data 2\n");

printf("(a) Write a recursive function to mirror\n");

printf("Before mirror:\n");

print(recur1, recur1[0]);

recursive(recur1, 1);

printf("After mirror:\n");

print(recur1, recur1[0]);

printf("(b) Write a iterative function to mirror\n");

printf("Before mirror:\n");

print(iter1, iter1[0]);

iterative(iter1);

printf("After mirror:\n");

print(iter1, iter1[0]);

}

void iterative(int \*a)

{

int t[20];

int count = 0;

int front;

int rear;

int time;

int j = 1;

for (int i = 0; i < 20; i++)

t[i] = power(2, i);

for (count = 0;; count++)

{

if (t[count] == a[0] + 1)

break;

}

count -= 1;

while (count--)

{

front = t[j];

rear = t[j + 1] - 1;

time = front / 2;

while (time--)

{

swap(&a[front], &a[rear]);

front++;

rear--;

}

j++;

}

}

void recursive(int \*a, int tochange)

{

int t[20];

for (int i = 0; i < 20; i++)

t[i] = power(2, i);

if (t[tochange] == a[0] + 1)

return;

int front = t[tochange];

int rear = t[tochange + 1] - 1;

int time = front / 2;

while (time--)

{

swap(&a[front], &a[rear]);

front++;

rear--;

}

recursive(a, tochange + 1);

}

int power(int x, int y)

{

int ans = 1;

while (y--)

{

ans \*= x;

}

return ans;

}

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

{

int temp;

temp = \*a;

\*a = \*b;

\*b = temp;

}

void print(int \*a, int time)

{

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

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

printf("\n");

}

Compilation:

gcc hw5\_3.c -o hw5\_3

Execution:

./hw5\_3

Output:

3.

test data 1

(a) Write a recursive function to mirror

Before mirror:

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

After mirror:

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

(b) Write a iterative function to mirror

Before mirror:

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

After mirror:

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

test data 2

(a) Write a recursive function to mirror

Before mirror:

31 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

After mirror:

31 1 3 2 7 6 5 4 15 14 13 12 11 10 9 8 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16

(b) Write a iterative function to mirror

Before mirror:

31 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

After mirror:

31 1 3 2 7 6 5 4 15 14 13 12 11 10 9 8 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16

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

Report: HW5\_4

Author: E94074029 江羿賢 <e94074029@gs.ncku.edu.tw>

Class: 資訊111級乙班

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

#include <stdio.h>

#include <stdlib.h>

void ip\_to\_bit(char \*);

void bit\_to\_ip(char \*);

int power(int, int);

int main()

{

//test data 1

char ip[16] = "129.160.96.1";

char bit[33] = "10000001101000000110000000000001";

printf("4.\n");

printf("test data 1 ip=129.160.96.1,bit=10000001101000000110000000000001\n");

printf("(a) print out its 32-bit bit pattern\n");

ip\_to\_bit(ip);

printf("\n");

printf("(b) print out it as IP address in the dot format\n");

bit\_to\_ip(bit);

//test data 2

char ip1[16] = "123.223.140.116";

char bit1[33] = "01111011110111111000110001110100";

printf("\n\n");

printf("test data 2 ip=123.223.140.116,bit=01111011110111111000110001110100\n");

printf("(a) print out its 32-bit bit pattern\n");

ip\_to\_bit(ip1);

printf("\n");

printf("(b) print out it as IP address in the dot format\n");

bit\_to\_ip(bit1);

printf("\n");

}

void ip\_to\_bit(char \*input)

{

char temp[4][4];

unsigned int n[4];

int store[4];

int row = 0;

int col = 0;

for (int i = 0; i < 16; i++)

{

if (input[i] != 0 && input[i] != 46)

{

temp[row][col] = input[i];

col++;

}

else

{

row++;

col = 0;

}

if (row == 4)

break;

}

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

{

store[i] = atoi(temp[i]);

n[i] = \*(unsigned int \*)&store[i];

}

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

{

for (int j = 7; j >=0; j--)

{

if ((n[i] >> j) & 1)

printf("1");

else

printf("0");

}

}

}

void bit\_to\_ip(char \*input)

{

int temp = 0;

int exp = 7;

for (int j = 0; j <= 24; j += 8)

{

temp = 0;

exp = 7;

for (int i = j; i < j + 8; i++, exp--)

{

if (input[i] == 49)

{

temp += power(2, exp);

}

}

printf("%d", temp);

if (j <= 16)

printf(".");

}

}

int power(int x, int y)

{

int ans = 1;

while (y--)

{

ans \*= x;

}

return ans;

}

Compilation:

gcc hw5\_4.c -o hw5\_4

Execution:

./hw5\_4

Output:

4.

test data 1 ip=129.160.96.1,bit=10000001101000000110000000000001

(a) print out its 32-bit bit pattern

10000001101000000110000000000001

(b) print out it as IP address in the dot format

129.160.96.1

test data 2 ip=123.223.140.116,bit=01111011110111111000110001110100

(a) print out its 32-bit bit pattern

01111011110111111000110001110100

(b) print out it as IP address in the dot format

123.223.140.116

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

Report: HW5\_5

Author: E94074029 江羿賢 <e94074029@gs.ncku.edu.tw>

Class: 資訊111級乙班

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

#include <stdio.h>

void boolfunc(int \*A, int n, int k);

void recursivebool(int \*A, int n, int k, int choosed, int putin);

int combinations(int \*A, int n, int k);

int temp[50];

int main()

{

printf("5.\n");

int A[4] = {4, 1, 2, 3};

int n = 4;

int k = 2;

printf("test data 1 A[4] = {4, 1, 2, 3},n = 4,k = 2\n");

combinations(A, n, k);

printf("\n");

int A1[5] = {4, 1, 2, 3, 5};

int n1 = 5;

int k1 = 3;

printf("test data 2 A[5] = {4, 1, 2, 3, 5},n = 5,k = 3\n");

combinations(A1, n1, k1);

}

void boolfunc(int \*A, int n, int k)

{

for (int i = 0; i < k - 1; i++)

{

if (temp[i] > temp[i + 1])

return;

printf("(");

for (int j = 0; j < k; j++)

{

if (j != k - 1)

printf("%d,", temp[j]);

else

printf("%d", temp[j]);

}

printf(") ");

return;

}

}

void recursivebool(int \*A, int n, int k, int choosed, int putin)

{

if (putin == k)

{

boolfunc(A, n, k);

return;

}

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

{

if (n - i < k - putin)

break;

temp[putin] = A[i];

recursivebool(A, n, k, i + 1, putin + 1);

}

}

int combinations(int \*A, int n, int k)

{

recursivebool(A, n, k, 0, 0);

printf("\n");

return 0;

}

Compilation:

gcc hw5\_5.c -o hw5\_5

Execution:

./hw5\_5

Output:

5.

test data 1 A[4] = {4, 1, 2, 3},n = 4,k = 2

(1,2) (1,3) (2,3)

test data 2 A[5] = {4, 1, 2, 3, 5},n = 5,k = 3

(1,2,3) (1,2,5) (1,3,5) (2,3,5)

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

Report: HW5\_6

Author: E94074029 江羿賢 <e94074029@gs.ncku.edu.tw>

Class: 資訊111級乙班

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

#include <stdio.h>

#include <stdlib.h>

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;

}

}

}

}

/\*for (t = 1, pr = 1; t <= (1 + j); t++)

pr = (-1) \* pr;\*/

//simplify to this

pr = (j % 2) ? 1 : -1;

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

}

for (j = 1, d = 0; j <= x; j++)

d += (f[1][j] \* c[j]);

return (d);

}

int main()

{

printf("6.\n");

printf("have changed the function determinant in hw5\_6.c\n");

printf("simplify line 14 \nfrom\n");

printf("for(t=1,pr=1;t<=(1+j);t++) pr=(-1)\*pr;\nto\n");

printf("pr = (j %% 2) ? 1 : -1;\n\n");

int pr = 1;

int t = 1;

printf("test the pr value by original line 14 that j 0~10\n");

for (int j = 0; j <= 10;j++)

{

for (t = 1, pr = 1; t <= (1 + j); t++)

pr = (-1) \* pr;

printf("%d ", pr);

}

printf("\n\ntest the pr value by new line 14 that j 0~10\n");

pr = 1;//initialize pr to 1 to do next test

for (int j = 0; j <= 10;j++)

{

pr = (j % 2) ? 1 : -1;

printf("%d ", pr);

}

printf("\n\nthese are equivalence!\nthe value of pr is same\n");

}

Compilation:

gcc hw5\_6.c -o hw5\_6

Execution:

./hw5\_6

Output:

6.

have changed the function determinant in hw5\_6.c

simplify line 14

from

for(t=1,pr=1;t<=(1+j);t++) pr=(-1)\*pr;

to

pr = (j % 2) ? 1 : -1;

test the pr value by original line 14 that j 0~10

-1 1 -1 1 -1 1 -1 1 -1 1 -1

test the pr value by new line 14 that j 0~10

-1 1 -1 1 -1 1 -1 1 -1 1 -1

these are equivalence!

the value of pr is same