201533661 이승수’s lab11/26

<lab1>

Step1: Understand the problem

Get the integer in 32-bit and named it Okay. Then reverse it on revOkay and print it.

Step2: Outline a solution

First, transfer Okay into bitsize word and reverse it by for loop.

Step3: Form a program structure

1. Get the 32-bit number and transfer it to bitsizing function.
2. Transfer it to bitsize and reverse it.
3. Print the revOkay.

Step4: Write a pseudo code

Void InitializeArr(int \*arr): put all the array elements 0;

Void Bittype(int num,int \*arr):make num into bitsize number;

Void main()

{

Get num;

Bittype(temp,Okay);

Print;

Reverse Okay;

Print Okay and revOkay;

}

Step5: Write the program

#pragma warning(disable.4996)

#include <stdio.h>

#include <math.h>

void InitializeArr(int arr[])

{

int i;

for (i = 0; i < 32; i++)

arr[i] = '0';

}

void Bittype(int num,int arr[])

{

int i=0,count=30;

printf("%d is<",num);

for (count = pow(2.0,30.0); count > 0; count=count/2)

{

arr[i] = num / count;

num = num %count;

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

i++;

}printf(">");

}

void main()

{

int Okay[33] = { '0' }, revOkay[33] = {'0'};

int temp,i;

InitializeArr(Okay);

InitializeArr(revOkay);

int d = pow(2.0, 4.0);

printf("Print is %d",d);

printf("Put number:");

scanf\_s("%d",&temp);

Bittype(temp,Okay);

printf("\nOkay: ");

for (i = 0; i < 31; i++)

{

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

}

for (i = 0; i < 32; i++)

{

revOkay[i] = Okay[31-i];

}

printf("\nrevOkay:");

for (i = 1; i < 32; i++)

{

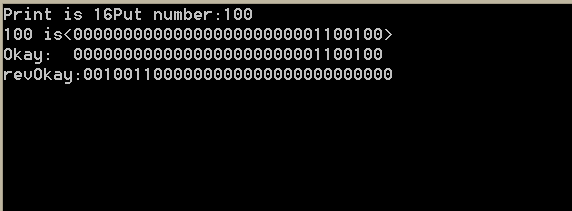
printf("%d", revOkay[i]);

}

getchar();

getchar();

}



<lab2>

Step1: Understand the problem

Get the strings from steno.txt and devide the string into digit-part and nondigit-part. Then store them on the txt file.

Step2: Outline a solution

Get the file and devide them by for looping and if conditions.

Step3: Form a program structure

1. Get steno.txt file and start reading.
2. At for loop, devide them by nondigit and digit part.
3. Write them on the nondigit.txt and digit.txt.

Step4: Write a pseudo code

inF=fopen(steno.txt)

while()

{

For(tempsize)

{

Devide into temp\_digit and temp\_Ndigit;

}

}

fclose(inF);

outF=fopen(digit.txt, nondigit.txt);

fprintf();

fclose(outF);

Step5: Write the program

#pragma warning(disable:4996)

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

void main()

{

FILE \*inF, \*outF;

int i,a=0,b=0;

char buff\_digit[1000] = { '\0', }, buff\_Ndigit[1000] = { '\0', }, temp[20] = { '\0', }, temp\_digit[20] = { '\0', }, temp\_Ndigit[20] = {'\0',};

inF = fopen("steno.txt", "r");

while (!feof(inF))

{

fscanf(inF,"%s",temp);

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

{

if ('0' <= temp[i] && temp[i] <= '9')

{

temp\_digit[a] = temp[i];

a++;

}

else

{

temp\_Ndigit[b] = temp[i];

b++;

}

}

strcat(buff\_digit, temp\_digit);

strcat(buff\_Ndigit, temp\_Ndigit);

a = 0;

b = 0;

strcpy(temp, "\0");//

strcpy(temp\_digit, "\0");//

strcpy(temp\_Ndigit, "\0");//

}

fclose(inF);

i = 0;

outF = fopen("digits.txt","w");

while (buff\_digit[i]!='\0')

{

fprintf(outF, "%c", buff\_digit[i]);

i++;

}

fclose(outF);

i = 0;

outF = fopen("nondigits.txt", "w");

while (buff\_Ndigit[i]!='\0')

{

fprintf(outF, "%c", buff\_Ndigit[i]);

i++;

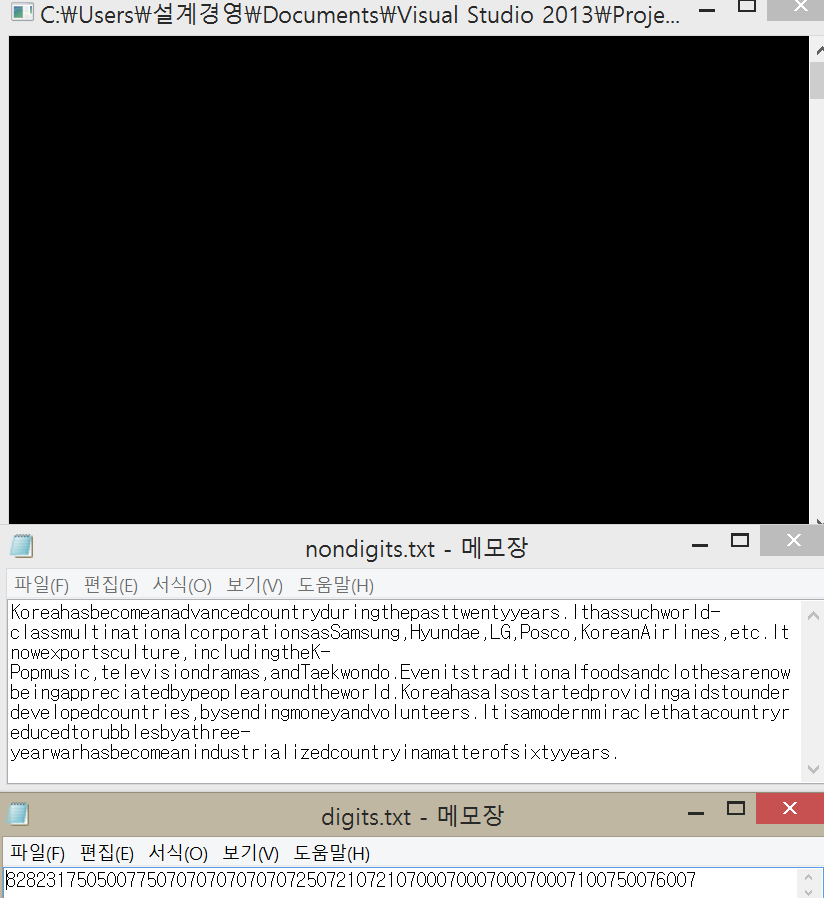
}

fclose(outF);

getchar();

getchar();

}



<lab3>

Step1: Understand the problem

Compute the time gap between two functions and draw them into the graph.

Step2: Outline a solution

Using the functions on the paper, compute the difference between tic and toc then compare them.

Step3: Form a program structure

1. Get the time gap between combination;
2. Get the time gap between tic and toc of combination2;
3. Print them on the program;
4. Show the differences by the graph;

Step4: Write a pseudo code

Long long combination(int n, int r);

Long long combination2(int n, int r);

Void main()

{

Tic-combination-toc;

Tic-combination2-toc;

Compute toc-tic of each combinations;

}

Show them on the graphs.

Step5: Write the program

#pragma warning(disable:4996)

#include <stdio.h>

#include <inttypes.h>

#include <time.h>

#include <Windows.h>

\_int64 GetMicroSecond();

long long combination(int n, int r)

{

if (r == 0 || n == r)

return 1;

return combination(n - 1, r - 1)

+ combination(n - 1, r);

}

#define MAXN 100

long long combination2(int n, int r)

{

static long long cache[MAXN][MAXN];

if (cache[n][r] > 0)

return cache[n][r];

if (r == 0 || n == r)

return (cache[n][r] = 1);

cache[n][r] = combination2(n - 1, r - 1) +

combination2(n - 1, r);

return cache[n][r];

}

void main()

{

int n = 3, r = 3;

\_int64 tic1 = 0, toc1 = 0,tic2=0,toc2=0;

tic1 = clock();

long long combination(n,r);

toc1 = clock();

tic2 = clock();

long long combination2(n,r);

toc2 = clock();

printf("%l64 1st microseconds\n",(toc1-tic1));

printf("%l64 2nd microseconds\n", (toc2 - tic2));

getchar();

getchar();

}

