```
************************************************************
Report: hw7
Author: F74046022 陳冠仁 <jeremy851004@gmail.com>
Class: 乙班
Description:
載入buckets.in並傳換成陣列bi[][]
以bi[][]得出要輸出至buckets.out的陣列bo[][]
以bi[][]、bo[][]得出要輸出至mapping.out的陣列map[][]
分別輸出bo[][]、map[][]至buckets.out、mapping.out

************************************************************
Code:
#include <stdio.h>

int i, j, x, y;
//loop variables
char ch;
//temporary char

int main(int argc, char *argv[])
{
	FILE *bc_in, *bc_out, *map_out;
	bc_in=fopen("/home/data/hw7/buckets.in", "r");
	bc_out=fopen("buckets.out", "w");
	map_out=fopen("mapping.out", "w");
	if(!bc_in)
	{
		printf("ERROR\n");
		return 0;
	}

	int line, size;
	line_size(bc_in, &line, &size);
	//line -> number of lines in buckets.in
	//size -> maximum number of numbers in a line

	int n=input_n(size);
	if(n<size)
```

```
	{
		printf("ERROR\n");
		return 0;
	}
	//input n (>=size)

	int bi[line][n], bo[line][n], map[line][n];
	int elm[line], idx[line], idx_max;
	//bi[][] -> array for buckets.in
	//bo[][] -> array for buckets.out
	//map[][] -> array for mapping.out
	//elm[] -> record how many numbers in each line
	//idx[] -> record the bucket index of each line
	//idx_max -> the maximum number of bucket used

	load_bi(bc_in, line, n, elm, bi);
	//transform buckets.in into bi[][]

	fill_bo(line, n, &idx_max, elm, bi, bo);
	//use bi[][] to generate bo[][]

	gnr_map(line, n, elm, bi, bo, map, idx);
	//use bi[][], bo[][] to generate map[][], idx[]

	output_bo(bc_out, n, idx_max, bo);
	output_map(map_out, line, n, map, idx);

	printf("TOTAL NUMBER OF BUCKETS : %d\n\n", idx_max);

	fclose(bc_in);
	fclose(bc_out);
	fclose(map_out);
	return 0;
}

int line_size(FILE *f, int *line, int *size)
{
	*line=1;
```

```
	*size=0;
	int cnt=1;

	while(fscanf(f, "%c", &ch)!=EOF)
	{
		if(ch==' ') cnt++;
		if(ch=='\n')
		{
			*line=*line+1;
			if(cnt>*size) *size=cnt;
			cnt=1;
		}
	}
	if(cnt>*size) *size=cnt;

	return;
}

int input_n(int size)
{
	int a;
	printf("\nPLEASE ENTER N (>=%d) : ", size);
	scanf("%d", &a);
	return a;
}

int load_bi(FILE *f, int line, int n, int elm[], int bi[][n])
{
	for(i=0; i<line; i++)
		elm[i]=1;
	//initialize elm[]

	i=0;
	rewind(f);
	while(fscanf(f, "%c", &ch)!=EOF)
	{
		if(ch==' ') elm[i]++;
		if(ch=='\n') i++;
```

```
	}
	//count how many numbers in each line

	rewind(f);
	for(i=0; i<line; i++)
		for(j=0; j<elm[i]; j++)
			fscanf(f, "%d", &bi[i][j]);
	//load buckets.in into bi[][]

	return;
}

int fill_bo(int line, int n, int *idx_max, int elm[], int
bi[][n], int bo[][n])
{
	*idx_max=0;

	for(i=0; i<line; i++)
		for(j=0; j<n; j++)
			bo[i][j]=0;
	//initialize array for buckets.out

	int bi_rem[line], bo_rem[line];
	for(i=0; i<line; i++)
	{
		bi_rem[i]=elm[i];
		bo_rem[i]=n;
	}
	//bi_rem[] -> how many number needs to be put in
	//bo_rem[] -> the remaining spaces of buckets

	for(i=0; i<line; i++)
	{
		for(x=0; x<line; x++)
		{
			for(j=0; j<elm[i]; j++)
				for(y=0; y<n; y++)
					if(bi[i][j]==bo[x][y])
```

```
                                {
                                        bi_rem[i]--;
                                        break;
                                }
                        if(bi_rem[i]>bo_rem[x])
                                bi_rem[i]=elm[i];
                        else
                                break;
                }
                //check if bo[x][] can be used

                for(j=0; j<elm[i]; j++)
                        for(y=0; y<n; y++)
                        {
                                if(bi[i][j]==bo[x][y])
                                        break;
                                if(bo[x][y]==0)
                                {
                                        bo[x][y]=bi[i][j];
                                        bo_rem[x]--;
                                        break;
                                }
                        }
                //fill bo[x][] with bi[i][]

                if(x+1>*idx_max) *idx_max=x+1;
                //record the maximum index used
        }

        return;
}

int gnr_map(int line, int n, int elm[], int bi[][n], int
bo[][n], int map[][n], int idx[line])
{
        for(i=0; i<line; i++)
                for(j=0; j<n; j++)
                        map[i][j]=0;
```

```
	//initialize the array for mapping.out

	int temp;
	//variable for checking

	for(i=0; i<line; i++)
	{
		for(x=0; x<line; x++)
		{
			temp=0;
			for(j=0; j<elm[i]; j++)
				for(y=0; y<n; y++)
					if(bi[i][j]==bo[x][y])
					{
						temp++;
						break;
					}
			if(temp==elm[i])
				break;
		}
		//check if bo[x][] can be used

		idx[i]=x;
		//mark the index of bucket

		for(j=0; j<elm[i]; j++)
			for(y=0; y<n; y++)
				if(bi[i][j]==bo[x][y])
				{
					map[i][y]=1;
					break;
				}
		//generate map[i][]according to bi[i][],
bo[x][]
	}

	return;
}
```

```
int output_bo(FILE *bc_out, int n, int idx_max, int bo[][n])
{
	for(i=0; i<idx_max; i++)
	{
		for(j=0; j<n; j++)
			fprintf(bc_out, "%d ", bo[i][j]);
		fprintf(bc_out, "\n");
	}
}

int output_map(FILE *map_out, int line, int n, int map[][n],
int idx[])
{
	for(i=0; i<line; i++)
	{
		fprintf(map_out, "%d ", idx[i]);
		for(j=0; j<n; j++)
			fprintf(map_out, "%d", map[i][j]);
		fprintf(map_out, "\n");
	}
}
```

Compilation:

```
gcc –o hw7 hw7.c
```

Execution:

```
./hw7
```

Output:

```
F74046022@c-2015-2:~/hw7> gcc -o hw7 hw7.c
F74046022@c-2015-2:~/hw7> ./hw7

PLEASE ENTER N (>=38) : 45
TOTAL NUMBER OF BUCKETS : 13

F74046022@c-2015-2:~/hw7> ./hw7_checker
no error
```

```
hw7.c: In function ‘line_size’:
hw7.c:66:2: error: incompatible type for argument 1 of
‘fscanf’
  while(fscanf(f, "%c", &ch)!=EOF)
  ^
In file included from /usr/include/features.h:364:0,
                 from /usr/include/stdio.h:27,
                 from hw7.c:1:
/usr/include/stdio.h:443:12: note: expected ‘struct FILE *
__restrict__’ but argument is of type ‘FILE’
 extern int __REDIRECT (fscanf, (FILE *__restrict __stream,
           ^


hw7.c:87:1: error: expected ‘;’ before ‘}’ token
 }
 ^


hw7.c:232:1: error: expected declaration or statement at end
of input
 }
 ^


hw7.c: In function ‘load_bi’:
hw7.c:96:2: error: incompatible type for argument 1 of
‘rewind’
  rewind(*f);
  ^
In file included from hw7.c:1:0:
/usr/include/stdio.h:759:13: note: expected ‘struct FILE *’
but argument is of type ‘FILE’
 extern void rewind (FILE *__stream);
             ^
```