<이산수학>_4장_프로그래밍 실습_C 코드

프로그래밍 실습 1

```
#include<Windows.h>
#define MAXITEM 10

void main()
{
	int m[MAXITEM+1][MAXITEM+1];
	int i,j,x,y,max;

	printf("입력으로 넣을 관계 행렬의 행의 크기는?₩n");
	scanf("%d", &max);
	printf("₩n");
	printf("1과 0으로 데이터를 입력하세요.?₩n");

	for(i=1;i<=max;i++)
	{
		for(j=1;j<=max;j++)
		{
			scanf("%d",&m[i][j]);
		}
	}
	printf("₩n");

	for(i=1;i<=max;i++)
	{
		for(j=1;j<=max;j++)
		{
			if(m[i][j] == 1)
			{
				for(x=1;x<=max;x++)
				{
					if(m[j][x] == 1)
					{
						for(y=1;y<=max;y++)
						{
							if(m[x][y] == 1)
							{
							    printf("(%2d%2d%2d%2d) => (%d..%d)₩n",i,j,x,y,i,y);
							}
						}
					}
				}
			}
		}
	}

	system("PAUSE");
}
```

프로그래밍 실습 2

```
#include<stdio.h>
#include<conio.h>
#include<Windows.h>

#define domain 20

void readfile(char fn[13],char adjancy[domain+1][domain+1]);

void main()
{
	int i,j,k,test;
	char fn[13] = "PP2-1.DAT";
	char adjancymx[domain+1][domain+1];
	char reflexity,symmetry,transitivity;

	for(i=1;i<=domain;i++)
	{
		for(j=1;j<=domain;j++)
		{
			adjancymx[i][j] = 0;
		}
	}

	/*printf("file name for read ? ");
	scanf("%s",fn);*/

	readfile(fn,adjancymx);

	printf("₩n*******************************₩n₩n");

	reflexity = 1;

	for(i=1;i<=domain;i++)
	{
		if(adjancymx[i][i] == 0)
		{
			reflexity = 0;
			break;
		}
	}

	if(reflexity == 1)
	{
		printf("%5cR is reflexive relation₩n",' ');
	}else
	{
		printf("%5cR is not reflexive relation₩n",' ');
	}

	symmetry = 1;
	test = 1;
	for(i=1;i<=domain;i++)
	{
		for(j=1;j<=domain;j++)
		{
			if(adjancymx[i][j] == 1 && adjancymx[j][i] == 0)
```

```
                {
                        symmetry = 0;
                        test = 0;
                        break;
                }
        }

        if(test == 0)
        {
                break;
        }
}

if(symmetry == 1)
{
        printf("%5cR is symmetric relation₩n", ' ');
}else
{
        printf("%5cR is not symmetric relation₩n", ' ');
}


transitivity = 1;
test = 1;

for(i=1;i<=domain;i++)
{
        for(j=1;j<=domain;j++)
        {
                if(adjancymx[i][j] == 1)
                {
                        for(k=1;k<=5;k++)
                        {
                                if(adjancymx[j][k] == 1 && adjancymx[i][k] == 0)
                                {
                                        transitivity = 0;
                                        test = 0;
                                        break;
                                }
                        }
                        if(test == 0)
                        {
                                break;
                        }
                }
        }
        if(test == 0)
        {
                break;
        }
}
if(transitivity == 1)
{
        printf("%5cR is transitive relation₩n",' ');
}else
{
        printf("%5cR is not transitive relation₩n",' ');
}
```

```
	system("PAUSE");
}

void readfile(char fn[13],char adjancy[domain+1][domain+1])
{
	FILE *fp;
	int x,y;

	fp = fopen(fn,"r");

	while(!feof(fp))
	{
		fscanf(fp,"%d %d",&x,&y);

		adjancy[x][y] = 1;
		printf("%2c(%2d, %2d)₩t",' ',x,y);
	}
	fclose(fp);
}
```

---

pp2-1.dat

```
1  1
1  3
2  2
2  4
2  5
3  1
3  3
3  5
4  2
4  4
4  7
5  2
5  3
5  5
5  9
6  6
6  15
7  4
7  7
7  20
8  8
8  9
9  5
9  8
9  9
10  10
10  11
11  10
11  11
12  12
12  17
13  13
13  15
14  14
15  6
15  13
15  15
16  16
17  12
17  17
18  18
19  19
20  7
20  20
```

프로그래밍 실습 3

```
#include<stdio.h>
#include<conio.h>
#include<windows.h>
#include<process.h>

void main()
{
	int i,j,k,m,n,s,t,domain,**Mrs;

	printf("₩n                    ### How many domain : ");
	scanf("%d",&domain);
	Mrs = (int**)malloc(sizeof(int)*domain);
	for(i=0;i<domain;i++)
	{
		*(Mrs+i) = (int*)malloc(sizeof(int)*domain);
	}
	for(i=0;i<domain;i++)
	{
		for(j=0;j<domain;j++)
		{
			(*(*(Mrs+i)+j)) = 0;
		}
	}

	printf("₩n## Input Relation R(1..n) ##₩n");
	printf("ex) 1 2 (to end : -1 –1)₩n");
	do
	{
		scanf("%d %d",&i,&j);
		if(i == -1)
		{
			break;
		}else
		{
			(*(*(Mrs+i-1)+j-1)) = 1;
		}
	}while(1);

	printf("₩n#### Relation Mr ####₩n");
	for(i=0;i<domain;i++)
	{
		for(j=0;j<domain;j++)
		{
			printf("%d ",*(*(Mrs+i)+j));
		}
		printf("₩n");
	}

	printf("₩t₩t₩t<<Press Any Key>>");
	getch();
	printf("₩n");
```

```
	for(i=0;i<domain;i++)
	{
		for(j=i+1;j<i+domain;j++)
		{
			for(k=i+1;k<i+domain;k++)
			{
				m = j%domain;
				n = k%domain;
				if(Mrs[i][n] == 1 && Mrs[m][i] == 1)
				{
					Mrs[m][n] = 1;
				}
			}
		}

		printf("#### W%d #### ₩n",i+1);
		for(s=0;s<domain;s++)
		{
			for(t=0;t<domain;t++)
			{
				printf("%d ",*(*(Mrs+s)+t));
			}
			printf("₩n");
		}
		printf("₩n");
		for(s=0;s<domain;s++)
		{
			for(t=0;t<domain;t++)
			{
				if(*(*(Mrs+s)+t) == 1)
				{
					printf("(%d, %d) ",s+1,t+1);
				}
			}
		}
		printf("₩n");
		printf("₩t₩t₩t<<PressAnyKey>>");
		getch();
		printf("₩n₩n");
	}

	for(i=0;i<domain;i++)
	{
		free(*(Mrs+i));
	}
	free(Mrs);
	system("PAUSE");
}
```