

Описание набора данных

```
#include <stdio.h>
#include <stdarg.h>
typedef int T;
#include "SQL.h"
struct Graf { Link_List S; int N; };
typedef struct Graf* Link_Graf;

void List_print(Link_List a) { Link_Item p = a->Front;
    while(p) { printf("%d ",Item_get_node(p)+1); p=Item_get_next(p); }
Link_Graf Graf_create(int n) { int i; Link_Graf p = (Link_Graf)calloc(1,sizeof(struct Graf));
    p->S=(Link_List)calloc(p->N=n,sizeof(struct List));
    for(i=0;i<n;i++) List_ini(&p->S[i]); return p; }
Link_List Graf_Get_node(Link_Graf a,T name) { return &a->S[name]; }
void Graf_Set_node(Link_Graf a,T name,...) { T v; va_list p; va_start(p,name);
    while(v=va_arg(p,T)) List_Insert_back(&a->S[name-1],v-1); va_end(p); }
void Graf_print(Link_Graf a) { int i; for(i=0;i<a->N;i++) { printf("%d: ",i+1);
    List_print(&a->S[i]); putchar('\n'); } }
```

Обход графа в глубину

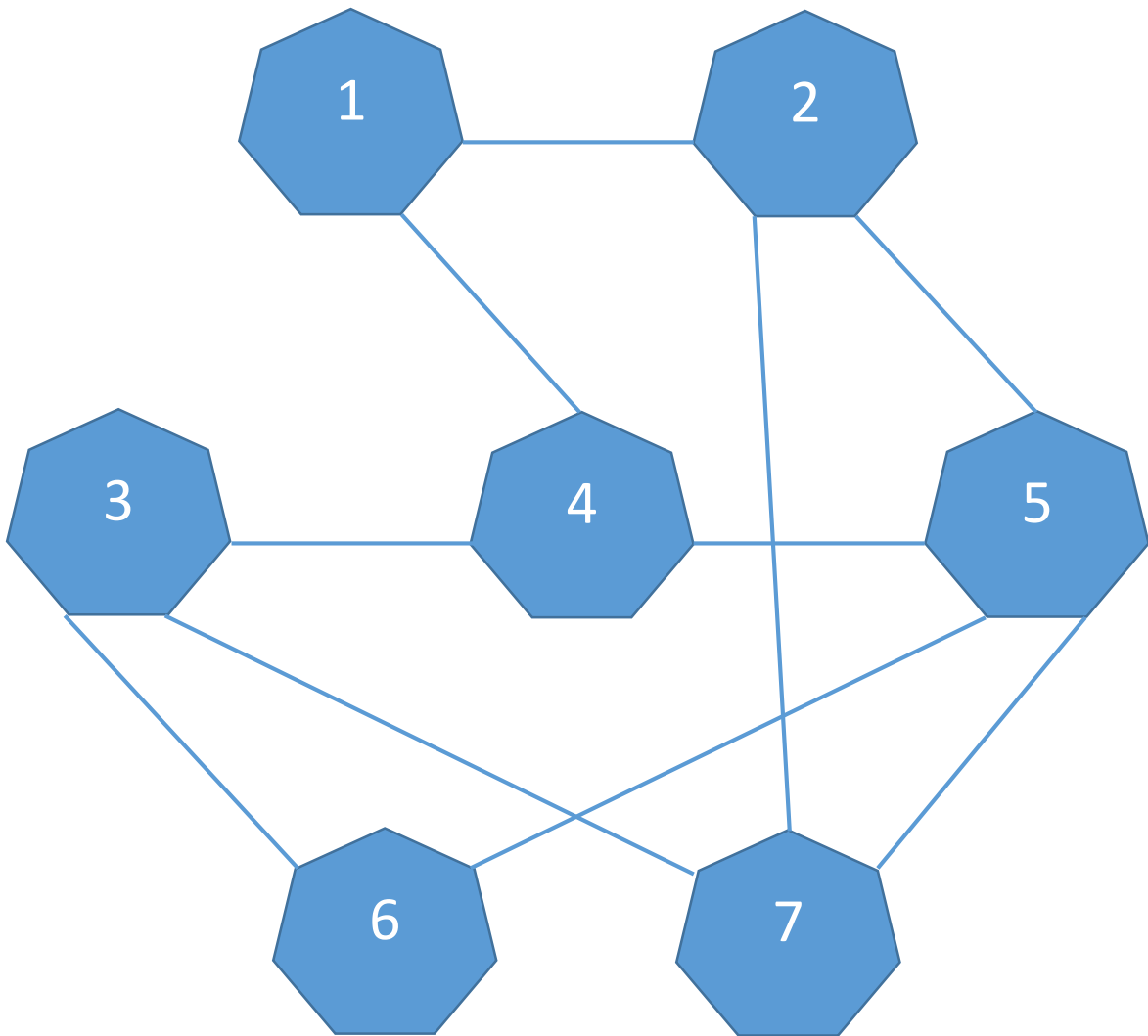
```
void print_V(Link_Graf A, int k) {
struct Stack Q; Stack_ini(&Q);
int *a = (int*)calloc(A->N,sizeof(int));
    Stack_push(&Q,k); a[k]++;
    while(!Stack_Is_Empty(&Q)) {
        while(!List_Is_Empty(Graf_Get_node(A,k))) {
            T i=List_Remove_front(Graf_Get_node(A,k));
                if(!a[i]) { a[i]++; Stack_push(&Q,i); printf("(%d,%d)\n",k+1,i+1); k=i;}
        }
        k=Stack_pop(&Q); } // end while
}
```

Обход графа в ширину

```
void print_G(Link_Graf A, int k) {
struct Queue Q; Queue_ini(&Q);
int *a = (int*)calloc(A->N,sizeof(int));
    Queue_put(&Q,k); a[k]++;
    while(!Queue_Is_Empty(&Q)) {
k=Queue_get(&Q);
while(!List_Is_Empty(Graf_Get_node(A,k)) {
    T i=List_Remove_front(Graf_Get_node(A,k));
    if(!a[i]) { Queue_put(&Q,i); a[i]++; printf("(%d,%d)\n",k+1,i+1); }
    }
    } // end while
}
```

Пример описания графа и построение путей обхода

```
int main() {
Link_Graf G = Graf_create(7), g = Graf_create(7);
    Graf_Set_node(G,1,2,4,0);      Graf_Set_node(g,1,2,4,0);
    Graf_Set_node(G,2,1,5,7,0);    Graf_Set_node(g,2,1,5,7,0);
    Graf_Set_node(G,3,4,6,7,0);    Graf_Set_node(g,3,4,6,7,0);
    Graf_Set_node(G,4,1,3,5,0);    Graf_Set_node(g,4,1,3,5,0);
    Graf_Set_node(G,5,2,4,6,7,0);  Graf_Set_node(g,5,2,4,6,7,0);
    Graf_Set_node(G,6,3,5,0);      Graf_Set_node(g,6,3,5,0);
    Graf_Set_node(G,7,2,3,5,0);    Graf_Set_node(g,7,2,3,5,0);
    Graf_print(G);
    print_V(G,0); putchar('\n'); print_G(g,0);
    return 0;
}
```



```
D:\rab\S1\2014\graf.exe
1: 2 4
2: 1 5 7
3: 4 6 7
4: 1 3 5
5: 2 4 6 7
6: 3 5
7: 2 3 5
(1,2)
(2,5)
(5,4)
(4,3)
(3,6)
(3,7)
(1,2)
(1,4)
(2,5)
(2,7)
(4,3)
(5,6)
Process returned 0 (0x0)   execution time : 1.094 s
Press any key to continue.
```