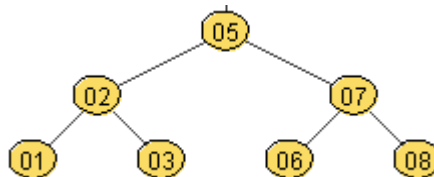


**ĐỀ THI KẾT THÚC MÔN CẤU TRÚC DỮ LIỆU**  
**Đề thi số 2 - Thời gian: 90 phút – Tổng điểm: 5 điểm**  
(Sinh viên **không** được phép sử dụng tài liệu, máy vi tính)

**Bài 1 (tối đa 3 điểm):**

- a. Trình bày công dụng của Stack và Queue? (0.5 điểm) Trình bày thuật toán duyệt cây theo chiều sâu sử dụng đệ quy 0.5 điểm)

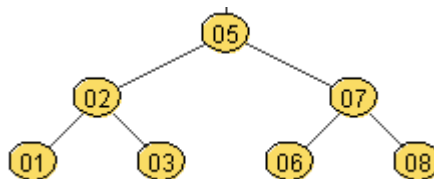
Ví dụ: ta có cây



Phép duyệt theo chiều sâu: 5, 2, 1, 3, 7, 6, 8

- b. Sinh viên thử sử dụng Stack để khử đệ quy trong thuật toán duyệt cây theo chiều sâu, viết đoạn chương trình trên. Giả thiết là chúng ta đã có sẵn các hàm của stack và queue (hàm *Init*, *Pop*, *Push*). (1 điểm)
- c. Sinh viên áp dụng Queue để viết đoạn chương trình tìm kiếm theo chiều rộng. Giả thiết là chúng ta đã có sẵn các hàm của stack và queue (hàm *Init*, *Pop*, *Push*). (1 điểm)

Ví dụ: ta có cây



Phép duyệt theo chiều rộng: 5, 2, 7, 1, 3, 6, 8

**Bài 2 (tối đa 3 điểm):**

Giả sử có danh sách liên kết (DSLK) có cấu trúc như sau:

```
typedef struct tagSINHVIEN
{
    int      MSSV;
    char     TenSV[100];
    float    DTB;
} SINHVIEN;
```

```
typedef struct tagNODE
{
    SINHVIEN      SV;
    struct tagNODE* pNext;
} NODE, *PNODE;
```

PNODE pHead; // là biến toàn cục, biến này quản lý danh sách liên kết

Viết hàm thực hiện việc:

1. Nhập từ bàn phím vào thông tin của 10 sinh viên. Thông tin này được đưa vào DSLK theo phương pháp **thêm vào cuối**. (0.5 điểm)
2. In ra danh sách những sinh viên có DTB < 5. (0.5 điểm)

3. Giải phóng danh sách liên kết (0.5 điểm)
4. Viết hàm main để gọi các hàm trên (0.5 điểm)

**Đại học quốc gia Tp HCM**  
**Trường Đại học Công nghệ Thông tin**

Cộng hòa xã hội chủ nghĩa Việt Nam  
Độc lập - Tự do - Hạnh phúc

**ĐÁP ÁN ĐỀ THI LÝ THUYẾT MÔN MÔN CẤU TRÚC DỮ LIỆU 1**  
**Đề thi số 2**

**Bài 1 (3 điểm):**

a. LIFO, FIFO

b.

```
void DuyetSau(BITREE proot)
{
    if(proot!=NULL)
    {
        printf(" %d",proot->info);
        DuyetSau(proot->left);
        DuyetSau S(proot->right);
    }
}
```

c.

```
BITREE Stack[100];
int Num=0;
void Giua1(BITREE proot)
{
    BITREE p;
    Num=1;
    Stack[0]=proot;
    while (Num>0)
    {
        Num--;
        p=Stack[Num];
        printf(" %d",p->info);
        if (p->right!=NULL) Stack[Num++]=p->right;
        if (p->left!=NULL) Stack[Num++]=p->left;
    }
}
```

d.

```
BITREE Queue[100];
int Begin=0;
int End=0;
void Giua2(BITREE proot)
{
    BITREE p;
    Begin=0;
    End=1;
    Queue[Begin]=proot;
    while (End-Begin>0)
    {
        p=Queue[Begin];
        Begin++;
    }
}
```

```

        printf(" %d",p->info);
        if (p->left!=NULL) Queue[End++]=p->left;
        if (p->right!=NULL) Queue[End++]=p->right;
    }
}

```

## Bài 2 (2 điểm):

```

#include <stdio.h>
#include <conio.h>
#include <malloc.h>
#include <string.h>

typedef struct tagSINHVIEN
{
    int    MSSV;
    char   HoTen[100];
    float   DTB;
}SINHVIEN;

typedef struct tagNODE
{
    SINHVIEN    SV;
    struct tagNODE*  pNext;
}NODE, *PNODE;
PNODE pHead=NULL;

PNODE CreateNode()
{
    PNODE p;
    SINHVIEN t;
    printf("Nhap vao MSSV");scanf("%d",&t.MSSV);
    fflush(stdin);
    printf("Nhap vao Ten");gets(t.HoTen);
    printf("Nhap vao DTB"); scanf("%f",&t.DTB);
    p = (PNODE)malloc(sizeof(NODE));
    //p->SV = t;
    p->SV.MSSV = t.MSSV;
    p->SV.DTB = t.DTB;
    strcpy(p->SV.HoTen,t.HoTen);
    p->pNext = NULL;
    return p;
}

void AddTailHead(PNODE q)
{
    if (pHead==NULL)
    {
        pHead = q;
    }
    else
    {
        p = pHead;
        while (p->pNext==NULL) p=p->pNext;
        p->pNext =q;
    }
}

void Cau1()
{
    for (int i=0;i<3;i++)
    {
        AddHead(CreateNode());
    }
}

```

```

    }
}
void Cau2 ()
{
    PNODE p;
    p=pHead;
    while (p!=NULL)
    {
        if (p->SV.DTB>=5)
        {
            printf ("%d\t%s\t%f",p->SV.MSSV,p->SV.HoTen,p->SV.DTB) ;
        }
        p=p->pNext;
    }
}
void Cau3 ()
{
    PNODE p=pHead;
    while (pHead!=NULL)
    {
        p=pHead;
        pHead=pHead->pNext;
        free (p) ;
    }
}
main ()
{
    Cau1 () ;
    Cau2 () ;
    Cau3 () ;
    getch () ;
}

```