

Kỹ thuật lập trình tuần 06

Bài trên lớp:

- bai 01.** Xây dựng danh sách liên kết đơn chứa số nguyên.
- bai 02.** Thêm 1 node vào đầu danh sách
- bai 03.** Thêm 1 node vào cuối danh sách
- bai 04.** Thêm 1 node vào vị trí thứ k của danh sách
- bai 05.** Xóa node đầu danh sách
- bai 06.** Xóa node cuối danh sách
- bai 07.** Xóa node vị trí thứ k của danh sách.
- bai 08.** Kiểm tra danh sách có rỗng hay không

Bài tập về nhà:

- bai 01.** Hãy cho biết chương trình có lỗi hay không? Giải thích.

```
#include<stdio.h>

int main()
{
```

Kỹ thuật lập trình tuần 06

```
int a[] = {10, 20, 30, 40, 50};  
int j;  
for(j=0; j<5; j++)  
{  
    printf("%d\n", a);  
    a++;  
}  
return 0;  
}
```

bai 02. Hãy cho biết chương trình có lỗi hay không? Giải thích.

```
#include<stdio.h>  
  
int main()  
{  
    int *x;  
    *x=100;  
    return 0;  
}
```

Kỹ thuật lập trình tuần 06

bai 03. Hãy cho biết cách giải phóng vùng nhớ của con trỏ.

```
#include<stdio.h>
#include<stdlib.h>
#define MAXROW 3
#define MAXCOL 4

int main()
{
    int **p, i, j;
    p = (int **) malloc(MAXROW * sizeof(int*));
    return 0;
}
```

bai 04. Hãy cho biết chương trình sau in ra màn hình cái gì? Giải thích.

```
#include<stdio.h>
#include<stdlib.h>

int main()
{
```

Kỹ thuật lập trình tuần 06

```
int *p;  
p = (int *)malloc(20); /* Assume p has address of 1314 */  
free(p);  
printf("%u", p);  
return 0;  
}
```

bai 05. Hãy cho biết chương trình sau in ra màn hình cái gì? Giải thích.

```
#include<stdio.h>  
#include<stdlib.h>  
  
int main()  
{  
    int *p;  
    p = (int *)malloc(20);  
    printf("%d\n", sizeof(p));  
    free(p);  
    return 0;  
}
```

Kỹ thuật lập trình tuần 06

- bai 01)** Xóa bỏ tất cả các phần tử trùng lặp (chỉ giữ lại một giá trị trong các giá trị trùng lặp)
- bai 02)** Tách danh sách liên kết đơn thành hai danh sách liên kết đơn khác. Trong đó danh sách thứ nhất sẽ chứa các phần tử nằm ở các vị trí lẻ của danh sách liên kết đơn ban đầu và danh sách thứ hai là các phần tử nằm ở vị trí chẵn của danh sách liên kết đơn ban đầu.
- bai 03)** Viết hàm đảo ngược danh sách liên kết đơn, tức là phần tử cuối sau khi đảo ngược sẽ trở thành phần tử đầu tiên.
- bai 04)** Viết hàm sắp xếp danh sách tăng dần.
- bai 05)** Viết hàm sắp xếp danh sách giảm dần.
- bai 06)** Xây dựng danh sách liên kết kiểu dữ liệu biểu diễn thông tin điểm trong mặt phẳng. Hãy viết hàm thực hiện các yêu cầu sau:
- Nhập điểm
 - Xuất điểm theo định dạng (x, y)
 - Tính khoảng cách giữa hai điểm.
 - Tính khoảng cách giữa hai điểm theo phương Ox
 - Tính khoảng cách giữa hai điểm theo phương Oy
 - Tìm điểm đối xứng qua gốc tọa độ.

Kỹ thuật lập trình tuần 06

- g. Tìm điểm đối xứng qua trục hoành.
- h. Tìm điểm đối xứng qua trục tung.
- i. Tìm điểm đối xứng qua đường phân giác thứ nhất ($y=x$).
- j. Tìm điểm đối xứng qua đường phân giác thứ hai ($y=-x$).
- k. Kiểm tra điểm có thuộc phần tư thứ I không?
- l. Kiểm tra điểm có thuộc phần tư thứ II không?
- m. Kiểm tra điểm có thuộc phần tư thứ III không?
- n. Kiểm tra điểm có thuộc phần tư thứ IV không?
- o. Đếm số lượng điểm có hoành độ dương
- p. Tìm một điểm có tung độ lớn nhất trong danh sách
- q. Tìm một điểm trong danh sách gần gốc tọa độ nhất
- r. Tìm hai điểm gần nhau nhất trong danh sách.
- s. Tìm hai điểm xa nhau nhất trong danh sách.
- t. Tìm 1 tam giác độc lập. Tam giác độc lập nghĩa là các điểm còn lại trong danh sách không nằm trên tam giác (các điểm trong danh sách khác nhau).
- u. Tìm một đa giác lồi sao cho nó chứa tất cả các điểm trong danh sách (các điểm trong danh sách khác nhau).

Kỹ thuật lập trình tuần 06