

آموزش شماره دو - برنامه سازی پیشرفته

در خدمت شما دوستان عزیز هستیم با سری آموزش های برنامه سازی پیشرفته C++ & C است که تقدیم حضورتان می شود.

در این آموزش ما به مباحث متغیرهای پویا (آرایه پویا ، استفاده متغیر های پویا همراه با تابع) و آرایه ای از اشاره گر ها می پردازیم. شما دوستان می توانید برای دریافت بقیه آموزش ها به سایت ITport.ir مراجعه کنید. امیدوارم این آموزش برای شما مفید واقع شود. با من همراه باشید.

متغیرهای پویا:

یکی دیگر از کاربردهای اشاره گر ها در تعریف متغیر پویا است. با استفاده از اشاره گر ها می توان که هر زمان که لازم بود مقداری از فضا از حافظه را به متغیر تخصیص داد و هر وقت لازم نداشتیم فضای آن را آزاد کنیم.

دستور malloc :

به وسیله این دستور فضایی از نوع پویا را از حافظه می گیریم و داخل اشاره گر می ریزیم. نحوی استفاده از

دستور:

(اندازه) malloc (*نوع)*ptr=

ptr : اشاره گر است.

مثال:

```
int*p=(int*)malloc(sizeof(int));
```

حال اگر به این فضا دیگر احتیاج نداشته باشیم و بخواهیم این فضا را آزاد کنیم از دستور به نام Free

استفاده می کنیم ، داریم:

```
Free(نام اشاره گر);
```

نکته: هر دوی این دستورات توسط هدر فایل `stdlib.h` پشتیبانی می شود.

مثال:

برنامه بنویسید که با استفاده از متغیرهای پویا دو متغیر را ایجاد کرده و توسط کاربر مقدار دهی نماید سپس مجموع مربعات آنها را چاپ نماید.

```
include<iostream.h>
include<stdlib.h>
main(){
int*p=(int*)malloc(sizeof(int));
int*q=(int*)malloc(sizeof(int));
if(!p&&!q){ ★
cin>>*p>>*q;
int s=*p**p+*q**q;
cout<<s;
Free(p);
Free(q); }
}
```

★ مفهوم این دستور: اگر فضایی خالی در حافظه باشد.

آرایه پویا:

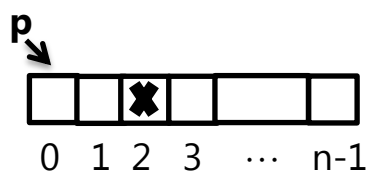
در آرایه معمولی، نام آرایه در واقع یک اشاره گر بوده ولی از نوع ایستا، اما آرایه پویا، آرایه ای است که هر موقع بخواهیم آن را ایجاد می کنیم و هر موقع بخواهیم آن را آزاد می کنیم، و می توانیم سائز آرایه را همانقدر که می خواهیم تعریف کنیم.

تعریف آرایه پویا:

به مثال توجه کنید:

```
int*p,n;
cin>>n;
p=(int*)malloc(sizeof(int)*n);
```

★ { * $(p+2)$ برای دسترسی به خانه‌های آرایه، یعنی به خانه 2 اشاره می‌کند.
p[2] و یا می‌توانیم بنویسیم



★ این دو معادل هم‌اند و می‌توانیم به محتوای خانه‌های آرایه دسترسی پیدا کنیم.

در آرایه معمولی به این صورت به خانه‌های آرایه دسترسی پیدا کنیم:

(اندیس+نام آرایه)*

مثال:

برنامه‌ای بنویسید که تعداد اعداد را از کاربر دریافت کرده سپس آن را از آخر به اول چاپ نماید

(تعداد اعداد توسط کار بر مشخص می‌شود)

چون تعداد اعداد نامشخص است بهتر است بریم سراغ آرایه پویا:

```
include<iostream.h>
include<stdlib.h>
main(){
int *p,n,i,j;
cin>>n;
```

```

p=(int*)malloc(sizeof(int)*n);
for(int i=0;i<n;i++)
    cin>>p[i];
for(int j=n;j>=0;j--)
    cout<<p[j];
}

```

استفاده متغیرهای پویا همراه با تابع:

یعنی یک لیست پویا را به یک تابع بفرستیم.

شکل دیگر فراخوانی با ارجاع: (به مثال توجه کنید)

```

main(){
int a,b;
cin>>a>>b;
swap(&a,&b); ★
cout<<a,b;
void swap(int*p,int*q) ★★
int temp;
temp=*p;
*p=*q;
*q=temp;
}

```

★ باید آدرسها را بفرستیم به تابع چون p و q اشاره گرند و a و b متغیراند.

★★ اگر بخواهیم اشاره گرها را به عنوان آرگومان مشخص کنیم باید این دستور را بنویسیم.

مثال:

برنامه‌ای بنویسید که n عدد را از کاربر دریافت کرده و در لیستی قرار دهد. سپس عنصر میانه لیست را چاپ نماید.

راهنمایی: یعنی باید تابعی بنویسیم که تابع عملیات مرتب سازی را انجام دهد. سپس مقدار میانه را چاپ نماید.

```

include<iostream.h>
include<stdlib.h>
main(){
int *p,n;
cin>>n;
p=(int*)malloc(sizeof(int)*n);
fill(p,n);    پر کردن لیست
sort(p,n);    مرتب سازی
n=display(p,n);
}

void fill(int *p,int n){
for(int i=0;i<n;i++)
    cin>>p[i];
}

void sort(int *p,int n){
int i,j,temp;
for(i=n;i>0;i--)
for(j=0;j<i;j++)
    if(p[i]>p[j+1]){
        temp=p[j];
        p[j]=p[j+1];
        p[j+1]=temp;
    }
}

int display(int*p,int n){
if(n%2)    یعنی اگر فرد باشد
return *(p+ $\frac{n}{2}$ );
return (*(p+ $\frac{n}{2}$ )+*(p+ $\frac{(n-1)}{2}$ ))/2;
}

```

تمرین:

تابعی را بنویسید که رشته‌ای را دریافت کرده و تمام حروف آن را به حروف بزرگ تبدیل نماید.
(طول رشته را نمی‌دانیم - باید از کد اسکی استفاده کنید).

نویسنده و مدرس: حسین جاهدی

