CSE CLUB
نادي هندسة الحاسوب

الأسبوع "الثالث عشر"

برمجة حاسوب "C++"

اسم الطالبة: سندس عوده

اسم الدكتور: د. محمد خليل

كتابة تطبيق (برنامج) يقوم بطباعة الاعداد الأولية (prime num)

```
bool prime (int x)
{
    for (int i=2 ; i<x/2 ; i++)
        if (x%2==0) return 0;

        return 1;
}

int main () {

int a[7] ;
for(int i=0 ; i<7 ; i++)
   cin>> a[i];

for (int i=0 ; i<7 ; i++)
   if ( prime (a[i])
   cout<< a[i] << " " ;

}
```

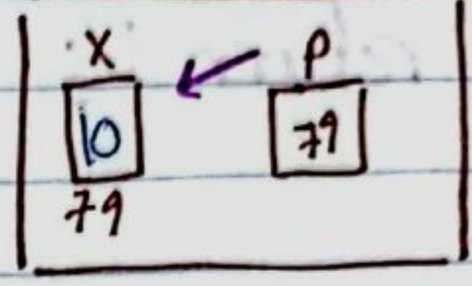
pointer

int *p ;
p = &a ;

يقوم البوينتر بحفظ العناوين لمتغيرات
من نفس نوع البوينتر

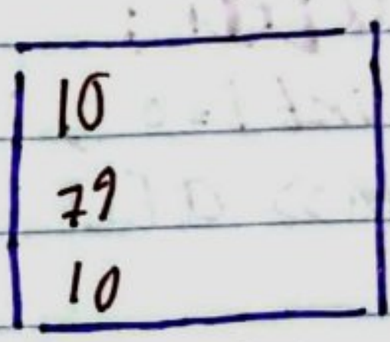
int x = 10 ;
int *p = &x ;

RAm



Cout << x <<endl ;
Cout << &x <<endl ;
Cout << *p <<endl ;
Cout << p <<endl ;

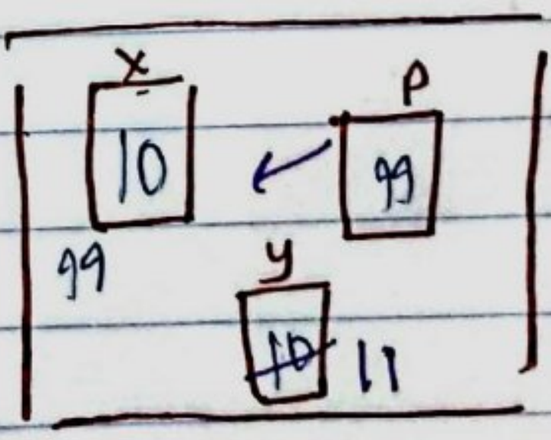
out put



* معناها إجلب القيمة الموجودة داخل الموقع الموجود (المخزن في p)
أي القيمة المخزنة في 79 وهي 10

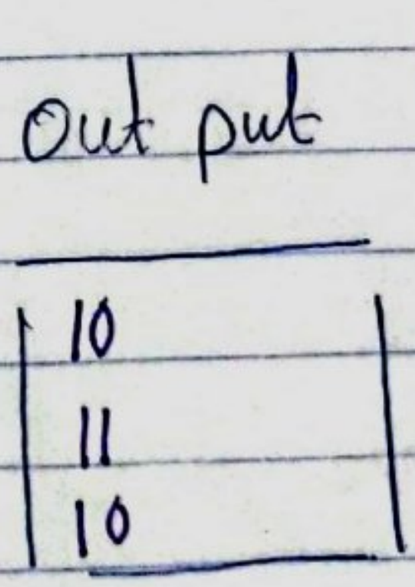
int a = 10 ;
int *p ;
p = &a ;

يمكن وضعهم في جملة
int *p = &a ;



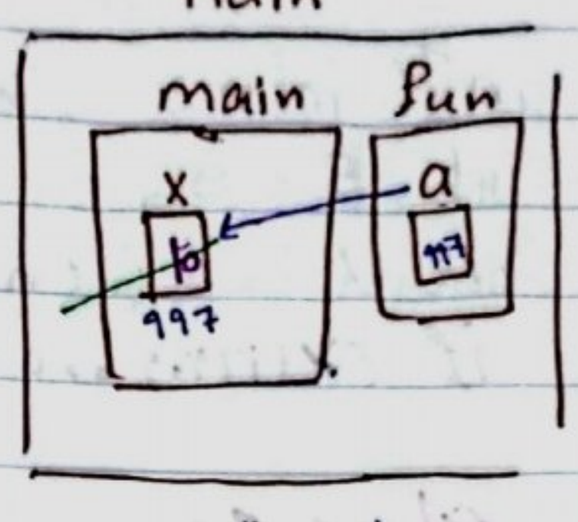
int y = *p ; y++ ;
Cout << "x = " << x << endl ;
Cout << "y = " << y << endl ;
Cout << "*p = " << *p << endl ;

out put



```
Void fun (int*a)
{
  *a = *a + 10;
}

int main() {
  int x = 10;
  Cout<< x <<endl;
  fun(&x);
  Cout << x <<endl;
```



output

10
20

ملاحظة :- نرسل الى الفنكشن اسم الـ array لانه يحمل موقع أول عنصر

ملاحظة :- اذا عندي عنوان (أي حد) بقدر أوصله يعني معي عنوان x فبقدر اعدل ع القيمة عن طريق التعديل ع البوينتر الذي معه عنوان x عند التعديل ع *a يتم التعديل ع x

```
void print (int* x, int n)   // (function print) لطباعة
{
   for(int i=0; i<n; i++)  cout<< x[i] <<" ";
}
void fun (int y[] ,int n)   // function لزيادة قيمة العناصر بواحد
{                           //
   for(int i=0; i<n; i++)  y[i]++ ;
}
int main() {
  int a[] = {10, 20, 30, 40, 7}
  print(a, 5);
  fun(a, 5);
  print(a, 5);
}
```

print(a, 5)
ارسلنا عنوان الـ array
وسعة الـ array عدد العناصر

output

10	20	30	40	7
11	21	31	41	8

ايجاد عدد الأعداد الزوجية والفردية داخل ال array

```
int FunE( int* x , int n )
{   int e = 0;
    for(int i = 0 ;i < n; i++)
      if (x[i]%2 == 0) e++;

    return e;
}
int FunO (int* x , int n)
{ int o = 0;
  for (int i=0 ;i < n ; i++)
        if(x[i] %2)  o++;
  return 0;
}
int main() {
  int a[] = {10 , 11, 30 , 5, 7 , 2 , 9};
  int n = 7;
cout << FunE (a , n ) << endl;
cout << FanO (a , n ) << endl;
}
```

out put →

3
4

فنكشن يقوم باستقبال array وعدد للبحث عنه داخلها (ويستقبل عدد الاعداد داخل الarray) اذا كان موجود يرجع قيمة الاندكس واذا لا يرجع -1

// linear search

```
int Fun (int *x , int n , int j)
{   for (int i=0 ;i<n ; i++)
      if (x[i] == j)
        return i;
    return -1;
}
```

فنجشن Swap لعددين داخل ال array

```
void Swap (int arr[], int ind1, int ind2)
{
    int T = arr[ind1];
  arr[ind1] = arr[ind2];
   arr[ind2] = T;
}
```

حوّش على ال array مباشرة

```
int main () {
int a[] = { 3, 8, 5, 1, 9};
   Swap ( a, 0, 4);
   for (int i=0; i< 5 ; i++)
    cout << a[i]<< " ";
}
```

output

| 9 8 5 1 3 |

Selection Sort

array

3 , 8 , 5 , 10 , 6 , 4 , 0

ت حل هو أصغر رقم اذا لا ابحث عن اصغر رقم اعمل swap بينه وبين هذا الرقم Indx = 0

0 , 8 , 5 , 10 , 6 , 4 , 3

0 , 3 , 5 , 10 , 6 , 4 , 8

0 , 3 , 4 , 10 , 6 , 5 , 8

هل في أصغر من هاد الرقم اذا لا نحل سواب بينهم

0 , 3 , 4 , 5 , 6 , 10 , 8

وهكذا مع كل ال array من indx = 1 حتى indx = n-1

0 , 3 , 4 , 5 , 10 , 6 , 8

0 , 3 , 4 , 5 , 6 , 8 , 10

اصبحت Sorted (رتبت)

```
void swap (int * arr, int ind1, int ind2)
{ int T = arr [ind1];
  arr [ind1] = arr [ind2];
  arr[ind2] = T;
}
Void selectionSort (int *a, int n)
{ for (int i=0; i<n-1; i++)
  {   int minIndx = i;
  for (int j =0; j<n-i-1 ; j++)
      if (a[j] < a[minIndx])
  minIndx = j;

  swap (a, i , minIndx);
  }
}
```

bubble Sort

```
void swap (int *a , int ind, int ind2)
  void bubbleSort (int arr[] , int n)
    {
      for (int i=0 ; i<n-1 ; i++)
      {
        for (int j=0; j<n-i-1 ; j++)
          if (a[j]> a[j+1])
              Swap (a, j ,j+1)
        }

Void swap (int *a , int ind , int ind2)
  { int T = a[ind];
    a[ind] = a[ind2];
    a[ind2] =T;
  }

int main() {
int a[] = {14, 22, 13, 8, 51, 1, 12 };
int n = sizeof(a[])/ sizeof (a[0]) ;
for (int i=0; i<n ; i++) Cout << a[i] << " ";
bubble Sort ( a , n)
Cout <<endl;
for (int i=0 ; i<n ; i++) Cout <<a[i] << " " ";
```

طريقة العمل

14 22 13 8 51 1 12

هل الذي بعدي أقل مني إذا نعم سواب إذا لا اسكب

14 (22) 13 8 51 1 12

14 13 22 8 51 1 12

14 13 8 22 (51) 1 12

14 13 8 22 1 51 12

14 13 8 22 1 51 12

13 14 8 22 1 12 51

13 8 14 1 22 12 51

13 14 8 22 1 12 51

13 8 14 1 22 12 51

13 8 14 1 12 22 51

8 13 14 1 12 22 51

8 13 1 14 12 22 51

8 13 1 12 14 22 51

8 1 13 12 14 22 51

8 1 12 13 14 22 51

1 8 12 13 14 22 51

sorted

كلها :- أصبحت مرتبة Sort —

output BubbleSort

قبل 14 22 13 8 51 1 12

بعد 1 8 12 13 14 22 51

Binary Search :-

لازم تكون ال array مرتبة Sorted

```
int a[] = {2, 3, 5, 11, 17, 23, 29};
int start = 0, end = 6 (n-1), mid;
```

هسا بشكل بسيط مثلاً عندي كتاب وبدي اوصل لـ الصفحة 300 بطريقة ال Binary بروح بمسك الكتاب من الوسط يعني مثلاً لو الكتاب 400 صفحة بفتح ع الصفحة 200 وبشوف هل الصفحة 300 بعدها او قبلها اذا بعدها بستثني البحث من اللي قبلها واذا قبلها بستثني البحث مثلاً بعدها هسا 300 بعدما بيحث في اللي بعد 200 وبلش 200 صفحة اي قبل وهكذا حتى اوصل اي الصفحة 300 ,

```
int Binary search (int *a ; int start ....)
{ while (start <= end)
  { mid = (start + end) / 2 ;

    if (a[mid == K)
      return mid;
    else if (a[mid < K)
      start = mid + 1 ;
    else
      end = mid - 1;
  }
  .... return -1;
}
```

ارجاع array به تابع

```
int * odd( int *a , int S )
  {  int c=0;

   for(int i=0 , i<S , i++)
     if (a[i] % 2 == 1)
        c++ ;

   int *arr = new int [c+1] ;
     arr [0] = c ;
   for(int i=0, i<S, i++)
   { if (a[i] %2 == 1 )

   arr  [i+1] = a[i];
   }

     return arr ;

   }

int main()
  {
int array[] = { 1, 4, 5, 9, 8 }
 int n = 5;

int *a =  odd(array , n );

 for (int i=1 ; i< a[0] ; i++)
    cout<< a[i] << " " ;
}
```

Two daimintional Array

Rows عدد الصفوف , Columns تتكون من أعمدة

Note

عدد الاعمدة
" يحدد الحاص " 3 كل صف

double mark [5] [4] ;

لانه عدد كسري ممكن يكون " عدد الـ 1d Array "

له عدد الصفوف

	0	1	2	3
	id	A	B	C
0	(0,0)	(0,1)	(0,2)	(0,3)
1	(1,0)	(1,1)	(1,2)	(1,3)
2	(2,0)	(2,1)	(2,2)	(2,3)
3	(3,0)	(3,1)	(3,2)	(3,3)
4	(4,0)	(4,1)	(4,2)	(4,3)

يمكن تعريفها بـ double ____ [.] [.]

data type نوع // array اسم // Rows // Columns

double arr [2] [2] = { 20, 40, 30, 50 }

الصف الثاني الصف الاول

int arr [2] [2] = { {20,3}, {1,53} }

float arr [1] [5] ;

int arr [1] [7] { } ; كلها أصفار

إدخال 2d Array

```
int a[2][3];
    for (int i=0, i<2; i++)
        for( int j=0; j<3; j++)
        cin>> a[i][j];
// cout << a[i][j] << " " ;   لطباعة الاصفار داخلها
```

تكون مرتبة في RAM بهذه الطريقة

(0,0)	(0,1)	(0,2)	(1,0)	(1,1)	(1,2)
1	1		1	1	

عند البحث عن العنصر نأخذ الصف والعمود

الصف هو عدد المصفوفات (الصفوف) اللازم تجاوزها للوصول الى المطلوب

العمود هو عدد الازاحات داخل المصفوفة المراد البحث فيها

مثلاً (1,0)

* عدد المصفوفات اللازم تجاوزها في الذاكرة 1
* عدد الازاحات اللازم هي صفر للوصول إلى العنصر

مثلاً (1,2)

* عدد المصفوفات اللازم تجاوزها هي 1 (المصفوفة 0) (الصف 0) "اول مصفوفة"
* عدد الازاحات اللازمة هي ازاحتين 2

وفي نهاية هذا التلخيص اتمنى التوفيق للجميع

اتمنى ان أكون قد أضفت إلى علمكم ❤

ختام آية الله تعالى الي طالما رافقتني

قال تعالى :« وقل ربِّ زدني علماً »

ختموا هذا البيت الشعري للإمام الشافعي
أذهب أنا بكم

بقدر الكد تكتسب المعالي

ومن طلب العلا سهر الليالي

ومن رام العلا من غير كدٍ

أضاع العمر في طلب المحالِ

بالتوفيق للجميع