

Subject:

Year. Month. Date. ()

```
void print Array (int a[ ][3])
```

```
{
```

```
    for (int i=0; i<2; i++) {
```

```
        for (int j=0; j<3; j++)
```

```
            cout << a[i][j] << " ";
```

```
        cout << endl;
```

```
    }
```

```
}
```

```
#include <iostream.h>
```

```
#include <iomanip.h>
```

```
const int students = 3;
```

```
const int exams = 4;
```

```
int minimum (int [ ][exams], int, int)
```

```
int maximum (int [ ][exams], int, int)
```

```
float average (int [ ], int);
```

```
void print Array (int [ ][exams], int, int)
```

```
int main ()
```

```
{
```

```
    int grades [students][exams] = { { 77, 68, 56, 73 }, { 90, 87, 89, 78 }, { 80, 90, 86, 81 } };
```

```
    cout << "grades : " << endl;
```

```
    print Array (grades, students, exams);
```

```
    cout << "minimum : " << minimum (grades, students, exams) << endl;
```

```
    cout << "maximum : " << maximum (grades, students, exams) << endl;
```

```
for (int person = 0; person < studentS; person++)  
    cout << "Average of student " << person << " is: " << setiosflags(ios::fixed) |  
        ios::showpoint << setprecision(2) << Average(grade[person],  
        exams) << endl;  
  
return 0;  
}
```

```
int minimum (int a[][exams], int n, int m)  
{  
    int min = 100;  
    for (int i = 0; i < n; i++)  
        for (int j = 0; j < m; j++)  
            if (a[i][j] < min)  
                min = a[i][j];  
    return min;  
}
```

```
float average (int a[], int n)  
{ int total = 0;  
    for (int i = 0; i < n; i++)  
        total += a[i];  
    return (float) total / n;  
}
```

فعلية

18 15

37 25

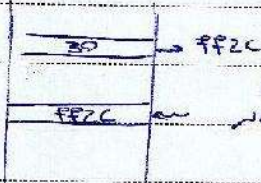
نسبي

Subject:

Year. Month. Date. ()

اساتذہ کرام

یہ فنکشن کسی ایسی فنکشن کے لیے لکھا گیا ہے جو کسی فنکشن کے پیرامیٹرز کو



یہ فنکشن کسی ایسی فنکشن کے لیے لکھا گیا ہے جو کسی فنکشن کے پیرامیٹرز کو

int *p

یہ فنکشن کسی ایسی فنکشن کے لیے لکھا گیا ہے جو کسی فنکشن کے پیرامیٹرز کو

یہ فنکشن کسی ایسی فنکشن کے لیے لکھا گیا ہے جو کسی فنکشن کے پیرامیٹرز کو

int a = 30

int *p = &a

یہ فنکشن کسی ایسی فنکشن کے لیے لکھا گیا ہے جو کسی فنکشن کے پیرامیٹرز کو

a = 7

یہ فنکشن کسی ایسی فنکشن کے لیے لکھا گیا ہے جو کسی فنکشن کے پیرامیٹرز کو

*p = a

*p = 7

یہ فنکشن کسی ایسی فنکشن کے لیے لکھا گیا ہے جو کسی فنکشن کے پیرامیٹرز کو

یہ فنکشن کسی ایسی فنکشن کے لیے لکھا گیا ہے جو کسی فنکشن کے پیرامیٹرز کو

int *a, b

&

int *p

*x = x

*x = x

```
#include <iostream.h>
```

```
int main ( )
```

```
{
```

```
int a;
```

```
int * aptr;
```

```
a = 7;
```

```
aptr = &a;
```

```
cout << "address of a : " << &a << "value of aptr : " << aptr << endl;
```

```
cout << "value of a : " << a << "value of *aptr : " << *aptr << endl;
```

```
cout << "& and * are inverse : &*aptr = " << &*aptr << endl;
```

```
" * &aptr = " << * &aptr << endl;
```

```
return 0;
```

```
}
```

```

const
int f(int *p)
{
    (*p)++;
    return 0;
}
    
```

این تابع همیشه مقدار 0 را برمیگرداند و در این تابع پ در هیچ نقطه‌ای تغییر نمی‌کند. فقط از عدد 5 که در خط اول تعریف شده استفاده می‌کند و تغییرش را در خط دوم به 6 می‌کند. اما چون این تابع به صورت const تعریف شده، پس در خط دوم نمی‌تواند تغییر دهد. پس خط دوم خطی است که خط قرمز است و خط اول خطی است که خط سبز است.

```

int main()
{
    int x = 5;
    cout << x; // 5
    f(&x);
    cout << x; // 6
    return 0;
}
    
```

مقدار x در خط اول 5 است و در خط دوم 6 می‌شود. چون f به &x می‌گیرد و در آنجا x را تغییر می‌دهد.

این در واقع از آنجا که در خط اول x را 5 تعریف کرده‌ایم و در خط دوم به f می‌دهیم و f هم مستقل می‌شود.

```

int *p
const int *p
    
```

در خط اول p یک اشاره‌گر است که می‌تواند به هر جایی اشاره کند. در خط دوم p یک اشاره‌گر است که فقط می‌تواند به یک مکان ثابت اشاره کند.

```

int * const p p = &x
const int * p p = &x
    
```

در خط اول p یک اشاره‌گر ثابت است که به &x اشاره می‌کند. در خط دوم p یک اشاره‌گر است که فقط می‌تواند به یک مکان ثابت اشاره کند و به &x اشاره می‌کند.

```

const int * const p
    
```

این یک اشاره‌گر ثابت است که فقط می‌تواند به یک مکان ثابت اشاره کند.

```

int a[10];
int * const a;
    
```

در خط اول a یک آرایه است. در خط دوم a یک اشاره‌گر ثابت است که به آرایه a اشاره می‌کند.

subject:

Year: Month: Date: ()

در یک آرایه می توان اعمال ریاضی انجام داد

```
int a[10];
```

```
int *p = a;
```

پسوند اول آرایه اشاره می کند

```
*p = 2;
```

خود آرایه را با مقدار 2 جایگزین می کند

```
p++;
```

پسوند اول آرایه اشاره می کند به خانه بعدی

```
(*p) = 5;
```

2bit جلوی آورد

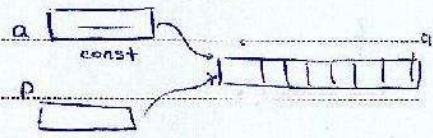
```
--p;
```

به عقب برگرداند

```
(*p) = 7;
```

```
p+3 = 3;
```

پسوند به صورتی بود
 $p+3 \equiv p[3] \equiv a[3]$



آرایه اشاره نمی کند (است که به آرایه اشاره می کند) const آرایه p به آرایه اشاره می کند
بلندیم ++p --p آرایه را تغییر می دهد

```
int *p1 = a;
```

آرایه اشاره می کند از هم که کرد

```
int *p2 = a + 5;
```

پسوند آرایه را 5 واحد جلو می برد

```
int x = p2 - p1;
```

تفاوت آرایه ها را می دهد (پسوند آرایه را از هم کم کرد)

```
void ToUpper(char s[])
```

"ali"

```
{ while (*s != '\0')
```

"ALI"

توی ToUpper در library وجود دارد (ctype.h)

```
{ *s += toupper(*s);
```

که حرف را به حرف بزرگ تبدیل می کند

```
s++;
```

```
}
```

Subject:

Year. Month. Date. ()

Type & size = size of

int a[10]

sizeof a = 10 * sizeof(int) = 20

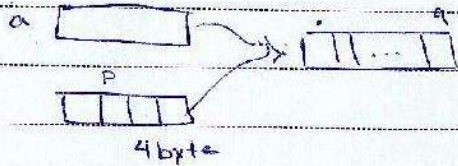
int *p = a;

a ≡ &a[0]

cout << sizeof(p);

0.0 = P (4 byte)

long int = 4 byte



به چه طریق می توانیم به این آدرس دسترسی کنیم

a[i] (آدرس)

*(a+i) (آدرس)

*(p+i)

const a در این خط است

const در این خط است

void copy1(char *s1, const char *s2)

s2 نمی تواند تغییر کند

```

{
    (s1[s1] = s2[s2]) != 0
    for (int i=0; s2[i] != 0; i++)
        s1[i] = s2[i];
}

```

این خط را می توانیم حذف کنیم

void copy2(char *s1, const char *s2)

```

{
    for (; (*s1 = *s2) != 0; s1++, s2++);
}

```

Subject:

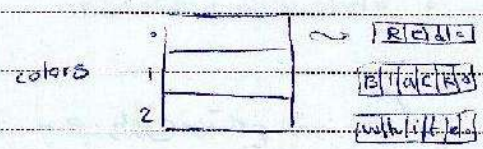
Year. Month. Date. ()

می توانیم برای این کار یک تابع بنویسیم که به ما بگوید که آیا یک عدد زوج است یا نه

```
int x = 10;
```



```
char x colors[] = {"Red", "Black", "white"};
```



```
for (int i = 0; i < 3; i++)
    cout << colors[i] << endl;
```

```
int f(int, int);
f(x, y);
```

فراخوانی

استاد به این سوال

دقت کنید با این کار که در این تابع

```
int (*p)(int, int);
```

توضیح: این تابع به ما میگوید که آیا یک عدد زوج است یا نه

integer و در این تابع integer می آید

پ = f
برای این نوع بازگشتی می توانیم از این استفاده کنیم

```
* p(x, y);
```

برای فراخوانی

این نوع بازگشتی می توانیم از این استفاده کنیم

Subject:

Year. Month. Date. ()

کامپوزیشن اور ترتیب (مثالاً)
#include <iostream.h>
#include <iomanip.h>

```
void bubble(int [], const int, int (*)(int, int));  
int ascending(int, int);  
int descending(int, int);
```

```
int main()  
{  
    const int size = 10;  
    int order, counter, a[size] = {2, 6, 4, 8, 10, 12, 87, 68, 45, 37};  
    cin << order;  
    cout << "Enter 1 to sort ascending, \n" << "Enter 2 to sort descending:";  
    cout << "Data in original order:" << endl;  
    for (counter = 0; counter < size; counter++)  
        cout << setw(4) << a[counter];  
  
    if (order == 1) {  
        bubble(a, size, ascending);  
        cout << "Data items in ascending order:\n";  
    } else {  
        bubble(a, size, descending);  
        cout << "Data in descending order:\n";  
    }  
}
```

Subject: _____

Year. _____ Month. _____ Date. _____

```
for (counter = 1; counter < size; counter++)  
    cout << setw(4) << a[counter];  
    cout << endl;  
return 0;  
}
```

```
void bubble (int b[], const int n, int (*compare)(int, int))  
{
```

```
    void swap (int *, int *)
```

```
    for (int pass = 1; pass < n; pass++)
```

```
        for (int i = 0; i < n - i; i++)
```

```
            if ((*compare)(b[i], b[i+1]))
```

```
                swap(&b[i], &b[i+1]);  
            }  
        }
```

→ true ~ b[i] > b[i+1], ascending
→ true ~ b[i] < b[i+1], descending

```
void swap (int * ep1, int * ep2)
```

```
{
```

```
    int temp = *ep1;
```

```
    *ep1 = *ep2;
```

```
    *ep2 = temp;
```

```
}
```

```
int ascending (int a, int b)
```

```
{ return a < b; }
```

```
int descending (int a, int b)
```

```
{ return a > b; }
```

```
P4PCO }
```

Subject: _____
 Year: _____ Month: _____ Date: _____

void (*f[3])(int) → *array of function pointers*
 size ↑
 integer
 void (*f[3])(int) → *array of function pointers*
 integer

void f1(int)	←
void f2(int)	←
void f3(int)	←

```
#include <iostream.h>
```

```
void f1(int);
```

```
void f2(int);
```

```
void f3(int);
```

```
int main()
```

```
{
```

```
void (*f[3])(int) = {f1, f2, f3};
```

```
int choice;
```

```
cout << "Enter your choice (1,2,3):";
```

```
cin >> choice;
```

```
while (choice >= 1 && choice <= 3)
```

```
{  
  (*f[choice-1])(choice);
```

```
  cout << "Enter your choice again (1,2,3 or another number to exit):";
```

```
  cin >> choice;
```

```
}
```

```
return 0;
```

```
}
```

Subject: _____

Year: _____ Month: _____ Date: _____

```
void f1(int a)
```

```
{
```

```
cout << "you selected " << a << " so f1 was called";
```

```
}
```

```
void f2(int a)
```

```
{
```

```
cout << "you selected " << a << " so f2 was called";
```

```
}
```

سوال (10) (10) soln
آرایه‌ها از char تا خوانده می‌شوند و در هر یک از آن‌ها کار می‌کنند

اگر حرف وارد کنند در cin space را ignore می‌کند در soln آن

بیش از تعداد حرف از خوانده می‌شود

اگر یک حرف بیشتر از تعداد حرف مورد نظر وارد کنند صاف جواب می‌دهد

cin یک C++ object است. getline دارد

white space را حذف می‌کند

```
cin.getline(s, 80, "\n");
```

حرف جدا کننده

اگر در این سطر خطی از cin می‌خواند و خطی از getline می‌خواند باید از این استفاده کرد

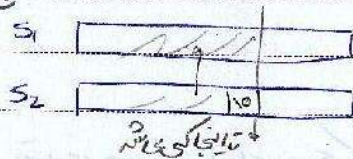
لازم است که در این سطر از 80 حرف یا بیشتر وارد شود و در این سطر باید از این استفاده کرد

بررسی نویسنده character رشته‌ها (string)

cin >> str → رشته مستقیم از ورودی خواند
 setw → چند حرف از زردی بخونه
 cin.getline → تا اندازه size که مشخص کردیم بخونه یا تا char خاصی که مشخص کردیم

<string.h>

* char * strcpy(char * S1, const char * S2) → کپی در یک رشته به رشته دیگر
 * در این سطر S1 مقادیر قبلی پاک می‌شود

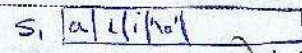


همه چیزی که در S2 در S1 کپی می‌شود
 S2 به صورت خودکار با 0 ختم می‌شود
 هیچ خطری ندارد برای حالتی که S1 ظرفیتش بیشتر از S2 باشد

```
char s1[10];
```

```
char s2[3] = "ali";
```

```
strcpy(s1, s2);
```



بقیه حافظه سطر خالی می‌ماند

*

```
char * strcpy(char * S1, const * S2, size_t n)
```

برای کپی کردن یک رشته به رشته دیگر با اندازه مشخص استفاده می‌شود
 چون محدودیتی در کپی کردن وجود دارد که از استفاده کنیم
 در این سطر S1 مقادیر قبلی پاک می‌شود
 در اینجا S2 به صورت خودکار با 0 ختم می‌شود
 S1 | a | l | i | \0 |
 S2 | a | l | i | \0 |
 strcpy(S1, S2)

```
typedef unsigned int uint
```

uint → unsigned int (سازماندهای کوچکتر از int)
 این نوع داده در برنامه‌های مختلف استفاده می‌شود

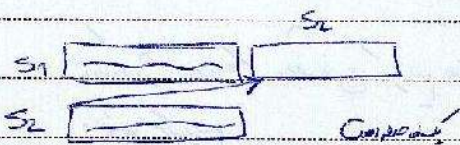
Subject:

Year. Month. Date. ()

* **توان کردن دو رشته به هم (جستار کردن)**

char * strcat (char * S1, const char * S2);

دو رشته را به هم می‌چسباند و در اولی جای می‌دهد. اگر در اولی جای خالی نباشد، در دوم به رشته اضافه می‌کند.



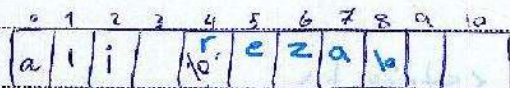
char S1[10] = "ali";

char S2[] = "reza";

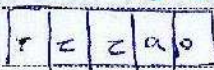
cout << S1;

strcat (S1, S2);

cout << S1;



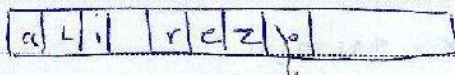
محل 10 را overwrite می‌کنند



بقایای حرف از ابتدای S2 که باقی ماند در انتهای S1

* char * strcat (char * S1; const char * S2, size_t n);

strcat (S1, S2, 3);



محدود این حرف خود در 3 تا از اولی می‌شود و 4 حرف اولی می‌ماند

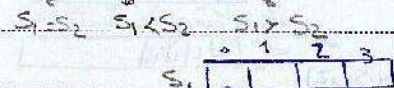
* **مقایسه دو رشته (بزرگتر یا کوچکتر بودن)**

int strcmp (const char * S1, const char * S2)

(=0) (<0) (>0)

(0 -1 +1) در صورتی که در اولی بزرگتر از دوم باشد

دو رشته مقایسه می‌شود و اگر در اولی بزرگتر از دوم باشد، عدد مثبت برمی‌گرداند. اگر کوچکتر باشد، عدد منفی برمی‌گرداند. اگر برابر باشند، 0 برمی‌گرداند.



خانه‌های 0 را به هم مقایسه می‌کنند. اگر مساوی بود، به خانه بعدی می‌روند.



اگر در (S1) < (S2) بود، S1 بزرگتر و اگر (S1) > (S2) بود، S2 بزرگتر است.

اگر در مقایسه مساوی بودند، به رشته بعدی می‌روند. اگر در رشته اولی به خالی رسیدند، عدد 0 برمی‌گرداند.

Subject: _____
 Year: _____ Month: _____ Date: _____

* `int strcmp(const char *s1, const char *s2, size_t n)` تعداد حرفی شده را مقایسه می کند

`s1` a|a|b|0

`if (strcmp(s1, s2) == 0)`

`s2` a|a|c|0

`cout << "equal by strcmp" << endl;`

`if (strcmp(s1, s2, 2) == 0)`

`cout << "equal by strcmp(s1, s2, 2)" << endl;`

* `while (s2[i++] != 0)` برای مقایسه طول استفاده می کند

`size_t strlen(const char *s)` تعداد کاراکترها

`strtok` رشته را به این تکه تکه می کند و در هر بار یک تکه را برمی گرداند

`char * strtok(char *s1, const char *s2)` date

`char date = "1386/01/27"` 09:20:36 space slash

`char * year, * month, * day` replace می کند و برمی گرداند

`year = strtok(date, " ")` space

`month = strtok(NULL, "/");` "/" نی

`day = strtok(NULL, ".");` "." نی

PAPCO `strtok` و `strtok` این دو تابع برای جدا کردن رشته ها استفاده می کنند

Subject:

Year. Month. Date. ()

رودها (class) *

③ struct record نامی و ساختار
پایه نوعی و ساختار نامی و ساختار

data نامی و ساختار نامی و ساختار (student) نامی و ساختار

```
struct student
```

```
{  
  int id;  
  char name[10];  
  ;  
}
```

data نامی و ساختار }
function method نامی و ساختار }
C++ class

Function نامی و ساختار Class نامی و ساختار struct

```
struct Time
```

```
{  
  int hour;  
  int minute;  
  int second;  
}
```

نامی و ساختار Type نامی و ساختار

Time t1, t2 نامی و ساختار integer type نامی و ساختار

Time * pt نامی و ساختار C++ نامی و ساختار

t1.hour = 3;

t1.minute = 20;

t1.second = 30;

Subject:

Year. Month. Date. ()

الترتیبی است. مقدار دهیم اما ابتدا برای
مقدار دهی استاندارد نیست

pt = 8 + 2;

pt -> hour = 3;

pt -> minute = 20;

pt -> second = 30;

برای دسترسی به این

(*pt).hour = 3;

(*pt).minute = 20;

(*pt).second = 30;

در این حالت بجای * از استاندارد نیست
(ا) اجباری است زیرا اولویت * از پرانتز بیشتر است و کما برکت
ی خواجه

```
# include <iostream.h>
```

```
struct Time
```

```
{
```

```
int hour;
```

```
int minute;
```

```
int second;
```

```
};
```

```
void printMilitary (const Time &t) {
```

```
void printStandard (const Time &t);
```

```
int main ()
```

```
{
```

```
Time dinnerTime;
```

```
dinnerTime.hour = 18;
```

```
dinnerTime.minute = 30;
```

```
dinnerTime.second = 0;
```

Subject

Year. Month. Date. ()

```
cout << "Dinner will be held at ";
```

```
printmilitary (dinnerTime);
```

```
print cout << "Military time, in which is ";
```

```
printstandard (dinnerTime);
```

```
cout << "standard time. " << endl;
```

```
dinnerTime.hour = 29;
```

```
dinnerTime.minute = 73;
```

```
cout << "Time with invalid values";
```

```
printmilitary (dinnerTime);
```

```
cout << endl;
```

```
return 0;
```

```
}
```

```
void printmilitary (const Time &t)
```

```
{
```

```
cout << t.hour < 10 ? "0" : "" << t.hour << ":
```

```
<< t.minute < 10 ? "0" : "" << t.minute << endl;
```

```
}
```

```
void printstandard (const Time &t)
```

```
{
```

```
cout << (t.hour == 0 || t.hour == 12 ? 12 : t.hour % 12)
```

```
<< ":" << (t.minute < 10 ? "0" : "") << t.minute
```

```
<< ":" << (t.second < 10 ? "0" : "") << t.second
```

```
<< (t.hour < 12 ? "AM" : "PM") << endl;
```

```
}
```

Subject:

Year. Month. Date. ()

class از زبان C++ می آید Type این نام مشخصه‌ای است برای شیء

class object کلاس
رده شیء

data members

class Time

اعضای داده‌ای

{

function members

public: → قابلیت دسترسی و به اشتراک

تابع عضو (method)

function members

Time (); t.setTime (5, 20, 30)

void setTime (int, int, int);

void printMilitary();

void printStandard();

private: → به خصوص تغییرات شیء و به اشتراک

data members

int hour; t.hour

int minute;

int second;

};

Time t1 → تابع برای تان تا زمانی که فراخوانی می‌شود

t1.setTime (6, 30, 20) ✓

t1.hour X

public و private هر دو می‌توانند در یک کلاس قرار بگیرند

برای اعضای داده‌ای و تابع عضو

Subject:

Year. Month. Date. ()

```
#include <iostream.h>
```

```
class Time
```

```
{
```

```
public:
```

```
Time ();
```

```
void setTime (int, int, int);
```

```
void printMilitary ();
```

```
void printStandard ();
```

```
private:
```

```
int hour;
```

```
int minute;
```

```
int second;
```

```
};
```

class در default, private

در هر دو دستاچ و دستایم رابطه با هم که در دستایم باشد
(سازنده - دستاچند ما هیچ که دستایم به دستایم نیستند)

در دستایم از دستایم که در دستایم به دستایم به دستایم
این دستاچ به دستایم است
فرضایم به دستایم

سازنده به دستایم که در دستایم به دستایم
فرضایم به دستایم
این دستاچ به دستایم است
فرضایم به دستایم

سازنده به دستایم که در دستایم به دستایم

```
Time t1 (6, 20, 30);
```

```
Time t2;
t2 = t1;
hour = minute = second = 0;
}
```