

```
char str1[] = "Hello"
```

```
char str2[] = "Hello"
```

```
if (str1 == str2)
```

```
    printf("Equal");
```

```
else
```

```
    printf("unequal");
```

str2	str1
'H'	'H'
'e'	'e'
'l'	'l'
'l'	'l'
'o'	'o'
'\0'	'\0'

هناك لنا char array و array z كل واحد منهما يحوي 7 بايت

وزي ما هنا عارفين ان اسم ال array هو address اول

عنصر في ال array ذو جيتا نقارنا بينه فيها من صيغتها

متساويين عشان كل واحد بيتساو على مكان مفتاح

→ output → unequal #

if the array begins at 1200 in memory

```
int arr[] = {2, 3, 4, 1, 6};
```

```
printf("%u, %u, %u", arr, &arr[0], &arr);
```

arr → اسم ال array هو address اول عنصر فيه

&arr[0] → ذا address اول عنصر في ال array

&arr → ذا address ال array كاملة

→ output → 1200, 1200, 1200

2	1200
3	1204
4	1208
1	1212
6	1216
	1220

↑ 4 byte

arr

ذا بيتساو على اول عنصر

ال array

&arr + 1

هيتنقل بمقدار 1 بايت الى ال array

في ال array

&arr

ذا بيتساو على ال array كاملة

&arr + 1

هيتنقل size ال array كامل وبيتساو

في ال array

What will happen if in a C program you assign a value to an array element whose subscript exceeds the size of array?

ex `int arr[5] = {1, 2, 3, 4, 5};`
`arr[5] = 10;`

arr[0]	1
arr[1]	2
arr[2]	3
arr[3]	4
arr[4]	5
arr[5] →	10

array size

هذا ال syntax error

Runtime error compilation error

عشان لما هتروح ز access على مكان ممكن يكون

فيه data انتا من برامجك ليا زينا

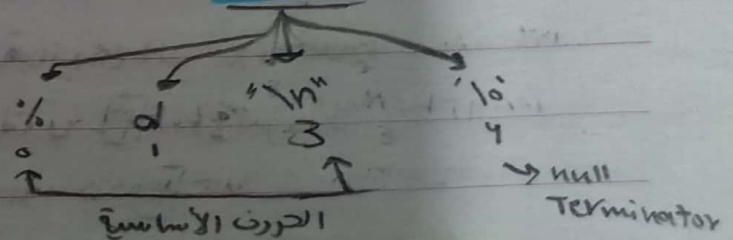
Output → The program may crash if some important data gets overwritten!

`char p[] = "%d\n";`
`p[1] = 'c';`
`printf(p, 65);`

ال p[] هذا ال size بتاعه

عبارة عن 4 بايت وهتبق
 مكونة من 3 حروف

`p[1] ⇒ d` متغير `c`



`p[] = "%c\n"`

`\n` → one character

`printf("%c\n", 65);`

char `65` زي ال 'A' وهو هنا متغير

Output → A


```
char arr[20] = "Mostafa"
```

sizeof(arr) → 20

لأن عدد عناصر ال array ب 20 وكل عنصر من نوع char

Total = 20 * 1 = 20 byte

strlen(arr) → 7

Function ال strlen ← يحس عدد حروف ال string

بعد ما يقابل أول (null terminator) "

```
char arr[7] = "Mostafa"
```

sizeof(arr) → 8

لأن هنا ال array متكون من 7 حروف

بنوع char مصغى + ال " " هيبقى الحرف الثامن

```
char arr[7] = "Mostafa"
```

strlen(arr) → 3

لأن هنا يقابل أول " " بعد حرف ال (5)

int → 2 byte

```
int arr[3][3] = { {1, 2}, {3, 4, 5}, {5} };
```

```
printf("%d %d %d", sizeof(arr), arr[0][2], arr[1][2]);
```

ال arr هنا هتبقى زي ال matrix

عدد ال columns هنا ب (3) وأسمه عدد

ال rows في ال arr ب (3)

→ sizeof(arr) = 3 * 3 * 2 = 18 byte

← هنا عننا في عناصر ناقصة فال Compiler (gcc) ب Zero

ب Zero

output → 18 0 5

1	2	0
3	4	5
5	0	0

(note) `int arr[5] = {1, 3, 5, 7, 9};`

`printf("%d", arr[3]);`

1
0
0
0
0

في الـ array، أول عناصرها هي 0 من نوع int وادينا قيمة
لأول عنصر فيها بس 1. خالص على الأول عناصر الـ array صاف
قيمة بـ zero [دك حاجة من الـ standard]

`arr[1] = arr[2] = arr[3] = arr[4] = 0`

Output → 0

`int arr[5];`

`printf("%d", arr[3]);`

Local

هنا مكنناش initialize لقيمة
واحدة على الأقل في الـ array وممكننا
هي local. كمان فالنتائج هتتغير

Garbage value

`int arr[5];`

`arr[3] → ??`

Global

هنا بردوا مكنناش initialize لقيمة
واحدة على الأقل في الـ array بس هنا الـ array
متعرفة كـ global بيتن ايتالي كل
القيمة بتاعها هتتغير بـ zero

`int a, b;`

`For (a=0, b=0; a<10, b<5; a++, b++)`

`printf("%d %d\n", a, b);`

عندنا الـ conditions مكون من جزئين $a < 10$ و $b < 5$ وبينهم
Comma والـ comma في الحالة دي separator. عينا ضاع
مع آخر condition عندنا الـ $(b < 5)$ و $a < 10$
صيطع عندنا القيم من صفر لـ 4

Output →

0 0

1 1

2 2

3 3

4 4

#

2. $str + 2 \rightarrow$ 1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47. 48. 49. 50. 51. 52. 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70. 71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84. 85. 86. 87. 88. 89. 90. 91. 92. 93. 94. 95. 96. 97. 98. 99. 100. 101. 102. 103. 104. 105. 106. 107. 108. 109. 110. 111. 112. 113. 114. 115. 116. 117. 118. 119. 120. 121. 122. 123. 124. 125. 126. 127. 128. 129. 130. 131. 132. 133. 134. 135. 136. 137. 138. 139. 140. 141. 142. 143. 144. 145. 146. 147. 148. 149. 150. 151. 152. 153. 154. 155. 156. 157. 158. 159. 160. 161. 162. 163. 164. 165. 166. 167. 168. 169. 170. 171. 172. 173. 174. 175. 176. 177. 178. 179. 180. 181. 182. 183. 184. 185. 186. 187. 188. 189. 190. 191. 192. 193. 194. 195. 196. 197. 198. 199. 200. 201. 202. 203. 204. 205. 206. 207. 208. 209. 210. 211. 212. 213. 214. 215. 216. 217. 218. 219. 220. 221. 222. 223. 224. 225. 226. 227. 228. 229. 230. 231. 232. 233. 234. 235. 236. 237. 238. 239. 240. 241. 242. 243. 244. 245. 246. 247. 248. 249. 250. 251. 252. 253. 254. 255. 256. 257. 258. 259. 260. 261. 262. 263. 264. 265. 266. 267. 268. 269. 270. 271. 272. 273. 274. 275. 276. 277. 278. 279. 280. 281. 282. 283. 284. 285. 286. 287. 288. 289. 290. 291. 292. 293. 294. 295. 296. 297. 298. 299. 300. 301. 302. 303. 304. 305. 306. 307. 308. 309. 310. 311. 312. 313. 314. 315. 316. 317. 318. 319. 320. 321. 322. 323. 324. 325. 326. 327. 328. 329. 330. 331. 332. 333. 334. 335. 336. 337. 338. 339. 340. 341. 342. 343. 344. 345. 346. 347. 348. 349. 350. 351. 352. 353. 354. 355. 356. 357. 358. 359. 360. 361. 362. 363. 364. 365. 366. 367. 368. 369. 370. 371. 372. 373. 374. 375. 376. 377. 378. 379. 380. 381. 382. 383. 384. 385. 386. 387. 388. 389. 390. 391. 392. 393. 394. 395. 396. 397. 398. 399. 400. 401. 402. 403. 404. 405. 406. 407. 408. 409. 410. 411. 412. 413. 414. 415. 416. 417. 418. 419. 420. 421. 422. 423. 424. 425. 426. 427. 428. 429. 430. 431. 432. 433. 434. 435. 436. 437. 438. 439. 440. 441. 442. 443. 444. 445. 446. 447. 448. 449. 450. 451. 452. 453. 454. 455. 456. 457. 458. 459. 460. 461. 462. 463. 464. 465. 466. 467. 468. 469. 470. 471. 472. 473. 474. 475. 476. 477. 478. 479. 480. 481. 482. 483. 484. 485. 486. 487. 488. 489. 490. 491. 492. 493. 494. 495. 496. 497. 498. 499. 500. 501. 502. 503. 504. 505. 506. 507. 508. 509. 510. 511. 512. 513. 514. 515. 516. 517. 518. 519. 520. 521. 522. 523. 524. 525. 526. 527. 528. 529. 530. 531. 532. 533. 534. 535. 536. 537. 538. 539. 540. 541. 542. 543. 544. 545. 546. 547. 548. 549. 550. 551. 552. 553. 554. 555. 556. 557. 558. 559. 560. 561. 562. 563. 564. 565. 566. 567. 568. 569. 570. 571. 572. 573. 574. 575. 576. 577. 578. 579. 580. 581. 582. 583. 584. 585. 586. 587. 588. 589. 590. 591. 592. 593. 594. 595. 596. 597. 598. 599. 600. 601. 602. 603. 604. 605. 606. 607. 608. 609. 610. 611. 612. 613. 614. 615. 616. 617. 618. 619. 620. 621. 622. 623. 624. 625. 626. 627. 628. 629. 630. 631. 632. 633. 634. 635. 636. 637. 638. 639. 640. 641. 642. 643. 644. 645. 646. 647. 648. 649. 650. 651. 652. 653. 654. 655. 656. 657. 658. 659. 660. 661. 662. 663. 664. 665. 666. 667. 668. 669. 670. 671. 672. 673. 674. 675. 676. 677. 678. 679. 680. 681. 682. 683. 684. 685. 686. 687. 688. 689. 690. 691. 692. 693. 694. 695. 696. 697. 698. 699. 700. 701. 702. 703. 704. 705. 706. 707. 708. 709. 710. 711. 712. 713. 714. 715. 716. 717. 718. 719. 720. 721. 722. 723. 724. 725. 726. 727. 728. 729. 730. 731. 732. 733. 734. 735. 736. 737. 738. 739. 740. 741. 742. 743. 744. 745. 746. 747. 748. 749. 750. 751. 752. 753. 754. 755. 756. 757. 758. 759. 760. 761. 762. 763. 764. 765. 766. 767. 768. 769. 770. 771. 772. 773. 774. 775. 776. 777. 778. 779. 780. 781. 782. 783. 784. 785. 786. 787. 788. 789. 790. 791. 792. 793. 794. 795. 796. 797. 798. 799. 800. 801. 802. 803. 804. 805. 806. 807. 808. 809. 810. 811. 812. 813. 814. 815. 816. 817. 818. 819. 820. 821. 822. 823. 824. 825. 826. 827. 828. 829. 830. 831. 832. 833. 834. 835. 836. 837. 83

صنا هينقل
بمقدار ٢٥
باب في الخطوة

Surrogate Value

driver's weight index[0] driver's

$\text{index} = \text{index} + 1$

أول ١٠ ، الثاني الناتج صيغ ← diaBix

```
int arr[size] = {1, 2, 3, 4, 5};
```

```
printf("%d", arr[2]);
```

← في الـ ANSI C ، لا يوجد compilation error

نظای ال size بتاع ال array بقيمة Variable حتى لو هو متغير

Constant: طبعیہ الحل ہوتا ہے مراحل میں بتا دے

Size ال عننا باستخدام ال Macros و ال

Java Macros inline error message

من قبل Pre Processor وهيئة استبدال ال size بالرقم

```
#define size 5
```

← **خبر** **ad** **و** **eg** **ح** **ارکلام** **بتای** **اننا** **نظمی** **اد** **size** **بتای**

Zusammenfassung: Constant Var., corr. 1!

```
int arr[1] = {10};
```

```
printf("%d\n", arr[0]);
```

out Put → 10

هستند روشها به شکل زیر منع غی ال Pointers

```
int i = 0;
```

```
int a[] = {1, 2, 5};
```

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

P → a

1
2
3

مثال

مکان تطبیق اریتمی ال array

→ arr[0] = 1

→ arr[0] = 1

→ p[0] = 1

→ *p = 1

→ *(p+0) = 1

→ *p + 0 = 1

دری مشورتی الی فوق

→ arr[2] = 5

→ arr[2] = 5

→ p[2] = 5

→ x x

→ *(p+2) = 5

→ *p + 2 = 1 + 2 = 3

هتجیه الی الاول *p و بعدین نهیه طیارا [2]

General

arr[i] = i[arr] = p[i] = *(p+i)

#

char data[2][3][2] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11};

data[a][b][c]

data[0][2][1] → 5

data[1][1][1] → 9

data[1][0][1] → 7

data[0][1][0] → 2

data[0][0][1] → 1

	(c) 0	1
(b) 0	0	1
(a) 0	1	2
	3	4
	5	6
	7	8
	9	10
	11	

(note)

double \rightarrow 8 byteint \rightarrow 2 bytechar \rightarrow 1 byte

The output =

long double a;

signed char b;

int arr[**sizeof(!a+b)**];

printf("%d", sizeof(arr));

كلما خرافات يا بنات شغالين و
عشان ال array متديش
error عندنا.

!a + b

zero \rightarrow a \rightarrow double

int

b

Absolute

قيمة

دولي

int + b

-b \rightarrow char \rightarrow int

نتقول

integer promotion

دا بيحصل لما بيكون عندنا متغير من نوع short و char

وبيحصل عملية حسابية مع int فالنتيجة صيغ

تلقاش ب int

char or short + int (or unsigned int) \rightarrow int

sizeof(!a+b) = sizeof(int) = 2 byte

int arr[2]

sizeof(arr) = 2 * 2

= 4 byte

(note)

! (any value) = 0

لوري
double float

دولي
int

Absolute int

```
char arr[11] = "The African Queen";
```

```
printf("%s", arr);
```

The African هنا هيحصل **Semantic error** ولما تطلع رن هيطلع

Null وبيد هالقيمة يطبع **garbage value** لحد ما يقابل أول

و ممكن في النص يحصل **Runtime error** و crash للبرنامج

فيها رد الفعل هيقت **undefined**

assume that **1022** is the memory address of "Hello1"

```
→ printf("%u", &"Hello1", &"Hello2");
```

address string
string

%u or %d
used to Print
Address

Output → **1022 Hello2**

array begins at **65472** int → 2 byte

```
int arr[3][4] = {1, 2, 3, 4, 4, 3, 2, 1, 7, 8, 9, 0}
```

```
printf("%u", a+1, &a+1);
```

2D Array

a → اسم ال array هنا هو address أول row

فعل ولما نرد كايها، قد فاحنا هنتحرك بقدر

8 = 4 * 2 ← row كامل

address(a+1) = 65472 + 8 = **65480** #

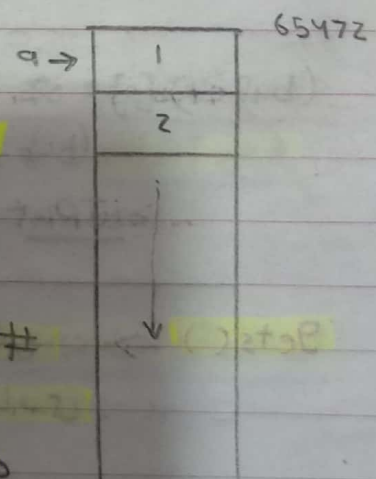
&a → دا بيستادر على ال array كاملة ولما

نتحرك نزيد نلوا حاجة ف هنتحرك بقدر ال array كلها

address(&a+1) = 65472 + 12 * 2

= **65496** #

Output → **65480, 65496**




```
char a1[3] = "abc";
char a2[8] = "abcdefgh";
char b1[] = "abc";
char b2[] = "abcdefgh";
```

Size of (a1) → 3 $3 * 1 = 3$

Size of (a2) → 8 $8 * 1 = 8$

Size of (b1) → 4 $3 * 1 + 1 = 4$ (1 → '0')

Size of (b2) → 9 $8 * 1 + 1 = 9$

char p;

char buf[10] = { 1, 2, 3, 4, 5, 6, 7, 8 };

p = (buf + 1)[5];

printf("%d", p);

buf[10] → { 1, 2, 3, 4, 5, 6, 7, 8 }

↑ buf+1

هنا نجد من أول هنا

(buf + 1)[0] = 2

(buf + 1)[5] = 9 #

output → 9

gets() → لما بنستخدمه وانا بنأخذ string فيش يتوقف على

ال size بتاع ال array و ممكن تأخذ أكبر منه كادى

وده ممكن يسبب مشكلة بعد كذا

ex char arr[5];

gets(arr); // Mostafa Edrees

printf("%s", arr);

output → Mostafa Edrees

`Pgets (arr-name, num of char, stdin)`

دى أحسن من `gets()` هنا بتقدر زحرد عدد الحروف ولو فى حروف ز يادها من بتدفعها ومن بتطبعها.

ex `char arr[5];`

`Pgets (arr, 5, stdin);` //Mostafa Edrees

`printf ("%s", arr);`

output → Mosta

صير طبعته ٤ حروف

بس عشان الحرف

الخامس هو الـ ٥

الى من يتقبل

بيقال string

note

`printf ("India", "Bix\n");`

هنا مفيش أقراس خال comma هيبه التعامل معاها

Separator وبالتالي صاخذ أول حاجة كذا

India

وبالتالى (النايج هيبه)

هنا موجود أقراس خال comma هيبه التعامل معاها

operator وبالتالي هنا خذ آخر حاجة كذا

`printf ("India", "Bix");` → output → Bix

note

عند التعامل مع الـ comma

لوفى أقراس ← بتبش operator هنا خذ آخر حاجة كذا

ex `int i = (1, 2, 3);`

`i = 3` ✓ #

لو مفيش أقراس ← بتبش separator وصاخذ أول حاجة كذا

ex `int i;`

`i = 1, 2, 3;`

`i = 1` ✓ #


```

char a[] = "\0" (note)
if (printf("%s", a)) ASCII of (\0) → 0
    printf("True");
else
    printf("False");

```

عندما نكتب string مكون من (\0) نكتبه Null terminator
والـ Null من بين عدد كبير وابتأى الـ printf يرجع Zero
يعني الـ condition بتأى الـ if بـ False يعني هتنتهي الـ else
والتأى الـ output هتكون False

```

char a = '\0'
if (printf("%c", a))
    printf("T");
else
    printf("F");

```

هذا الـ char هنا بـ \0
من string array of char

هنا الـ [0] هتد على إيه حرف واحد وبتأى الـ printf
هتدولنا واحد يعني الشرط بتأى بـ true يبقى هتدولنا
الـ body بتأى الـ if

output ⇒ space T

عشان بتدع الـ '\0' كـ char باستخدام (%c)

output ⇒ 0 T

في حالة لو طبعا استخدم (%d)

```
char a[] = "\0";
```

```
printf("%s", a);
```

output ⇒ (null)

```
printf("%d", a[0]);
```

output ⇒ 0 (zero)