[Tech Forum - Technical Discussion Forum](http://techforum4u.com/archive/index.php?s=b8c78db3c3d59180cf171a4da376f99b) > [Jobs And Career](http://techforum4u.com/archive/index.php/f-67.html?s=b8c78db3c3d59180cf171a4da376f99b) > [Career](http://techforum4u.com/archive/index.php/f-72.html?s=b8c78db3c3d59180cf171a4da376f99b) > [Technical Questions Papers and Discussions](http://techforum4u.com/archive/index.php/f-79.html?s=b8c78db3c3d59180cf171a4da376f99b) > Arrays in C

[PDA](http://techforum4u.com/archive/index.php/t-7579.html?s=b8c78db3c3d59180cf171a4da376f99b&pda=1)

View Full Version : [Arrays in C](http://techforum4u.com/showthread.php/7579-Arrays-in-C?s=b8c78db3c3d59180cf171a4da376f99b)

pakkiaraj kasinathan

12-14-2011, 06:14 AM

1.   
What will happen if in a C program you assign a value to an array element whose subscript exceeds the size of array?  
A. The element will be set to 0. B. The compiler would report an error. C. The program may crash if some important data gets overwritten. D. The array size would appropriately grow.  
Answer & Explanation  
Answer: Option C  
Explanation:  
If the index of the array size is exceeded, the program will crash. Hence "option c" is the correct answer. But the modern compilers will take care of this kind of errors.  
Example: Run the below program, it will crash in Windows (TurboC Compiler)  
#include<stdio.h> int main() { int arr[2]; arr[3]=10; printf("%d",arr[3]); return 0; }  
Since C is a compiler dependent language, it may give different outputs at different platforms. We have given the Turbo-C Compiler (Windows) output.  
Please try the above programs in Windows (Turbo-C Compiler) and Linux (GCC Compiler), you will understand the difference better.  
View Answer Online Compiler Report Discuss in Forum  
2.   
  
What does the following declaration mean?  
int (\*ptr)[10] A. ptr is array of pointers to 10 integers B. ptr is a pointer to an array of 10 integers C. ptr is an array of 10 integers D. ptr is an pointer to array  
Answer & Explanation  
Answer: Option B  
Explanation:  
No answer description available for this question. Let us discuss.  
View Answer Online Compiler Report Discuss in Forum  
3.   
In C, if you pass an array as an argument to a function, what actually gets passed?  
A. Value of elements in array B. First element of the array C. Base address of the array D. Address of the last element of array  
Answer & Explanation  
Answer: Option C  
1.   
What will be the output of the program ?  
#include<stdio.h> int main() { int a[5] = {5, 1, 15, 20, 25}; int i, j, m; i = ++a[1]; j = a[1]++; m = a[i++]; printf("%d, %d, %d", i, j, m); return 0; } A. 2, 1, 15 B. 1, 2, 5 C. 3, 2, 15 D. 2, 3, 20  
Answer & Explanation  
Answer: Option C  
Explanation:  
Step 1: int a[5] = {5, 1, 15, 20, 25}; The variable arr is declared as an integer array with a size of 5 and it is initialized to  
a[0] = 5, a[1] = 1, a[2] = 15, a[3] = 20, a[4] = 25 .  
Step 2: int i, j, m; The variable i,j,m are declared as an integer type.  
Step 3: i = ++a[1]; becomes i = ++1; Hence i = 2 and a[1] = 2  
Step 4: j = a[1]++; becomes j = 2++; Hence j = 2 and a[1] = 3.  
Step 5: m = a[i++]; becomes m = a[2]; Hence m = 15 and i is incremented by 1(i++ means 2++ so i=3)  
Step 6: printf("%d, %d, %d", i, j, m); It prints the value of the variables i, j, m  
Hence the output of the program is 3, 2, 15  
View Answer Online Compiler Report Discuss in Forum  
2.   
What will be the output of the program ?  
#include<stdio.h> int main() { static int a[2][2] = {1, 2, 3, 4}; int i, j; static int \*p[] = {(int\*)a, (int\*)a+1, (int\*)a+2}; for(i=0; i<2; i++) { for(j=0; j<2; j++) { printf("%d, %d, %d, %d\n", \*(\*(p+i)+j), \*(\*(j+p)+i), \*(\*(i+p)+j), \*(\*(p+j)+i)); } } return 0; }  
A. 1, 1, 1, 1  
2, 3, 2, 3  
3, 2, 3, 2  
4, 4, 4, 4 B. 1, 2, 1, 2  
2, 3, 2, 3  
3, 4, 3, 4  
4, 2, 4, 2 C. 1, 1, 1, 1  
2, 2, 2, 2  
2, 2, 2, 2  
3, 3, 3, 3 D. 1, 2, 3, 4  
2, 3, 4, 1  
3, 4, 1, 2  
4, 1, 2, 3  
Answer & Explanation  
Answer: Option C  
Explanation:  
No answer description available for this question. Let us discuss.  
View Answer Online Compiler Report Discuss in Forum  
3.   
What will be the output of the program ?  
#include<stdio.h> int main() { void fun(int, int[]); int arr[] = {1, 2, 3, 4}; int i; fun(4, arr); for(i=0; i<4; i++) printf("%d,", arr[i]); return 0; } void fun(int n, int arr[]) { int \*p=0; int i=0; while(i++ < n) p = &arr[i]; \*p=0; } A. 2, 3, 4, 5 B. 1, 2, 3, 4 C. 0, 1, 2, 3 D. 3, 2, 1 0  
Answer & Explanation  
Answer: Option B  
Explanation:  
Step 1: void fun(int, int[]); This prototype tells the compiler that the function fun() accepts one integer value and one array as an arguments and does not return anything.  
Step 2: int arr[] = {1, 2, 3, 4}; The variable a is declared as an integer array and it is initialized to  
a[0] = 1, a[1] = 2, a[2] = 3, a[3] = 4  
Step 3: int i; The variable i is declared as an integer type.  
Step 4: fun(4, arr); This function does not affect the output of the program. Let's skip this function.  
Step 5: for(i=0; i<4; i++) { printf("%d,", arr[i]); } The for loop runs untill the variable i is less than '4' and it prints the each value of array a.  
Hence the output of the program is 1,2,3,4  
View Answer Online Compiler Report Discuss in Forum  
4.   
What will be the output of the program ?  
#include<stdio.h> void fun(int \*\*p); int main() { int a[3][4] = {1, 2, 3, 4, 4, 3, 2, 8, 7, 8, 9, 0}; int \*ptr; ptr = &a[0][0]; fun(&ptr); return 0; } void fun(int \*\*p) { printf("%d\n", \*\*p); } A. 1 B. 2 C. 3 D. 4  
Answer & Explanation  
Answer: Option A  
Explanation:  
Step 1: int a[3][4] = {1, 2, 3, 4, 4, 3, 2, 8, 7, 8, 9, 0}; The variable a is declared as an multidimensional integer array with size of 3 rows 4 columns.  
Step 2: int \*ptr; The \*ptr is a integer pointer variable.  
Step 3: ptr = &a[0][0]; Here we are assigning the base address of the array ato the pointer variable \*ptr.  
Step 4: fun(&ptr); Now, the &ptr contains the base address of array a.  
Step 4: Inside the function fun(&ptr); The printf("%d\n", \*\*p); prints the value '1'.  
because the \*p contains the base address or the first element memory address of the array a (ie. a[0])  
\*\*p contains the value of \*p memory location (ie. a[0]=1).  
Hence the output of the program is '1'  
View Answer Online Compiler Report Discuss in Forum  
5.   
What will be the output of the program ?  
#include<stdio.h> int main() { static int arr[] = {0, 1, 2, 3, 4}; int \*p[] = {arr, arr+1, arr+2, arr+3, arr+4}; int \*\*ptr=p; ptr++; printf("%d, %d, %d\n", ptr-p, \*ptr-arr, \*\*ptr); \*ptr++; printf("%d, %d, %d\n", ptr-p, \*ptr-arr, \*\*ptr); \*++ptr; printf("%d, %d, %d\n", ptr-p, \*ptr-arr, \*\*ptr); ++\*ptr; printf("%d, %d, %d\n", ptr-p, \*ptr-arr, \*\*ptr); return 0; }  
A. 0, 0, 0  
1, 1, 1  
2, 2, 2  
3, 3, 3 B. 1, 1, 2  
2, 2, 3  
3, 3, 4  
4, 4, 1 C. 1, 1, 1  
2, 2, 2  
3, 3, 3  
3, 4, 4 D. 0, 1, 2  
1, 2, 3  
2, 3, 4  
3, 4, 5  
Answer & Explanation  
Answer: Option C  
1.   
Which of the following is correct way to define the function fun() in the below program?  
#include<stdio.h> int main() { int a[3][4]; fun(a); return 0; } A. void fun(int p[][4]) { } B. void fun(int \*p[4]) { } C. void fun(int \*p[][4]) { } D. void fun(int \*p[3][4]) { }  
Answer & Explanation  
Answer: Option A  
Explanation:  
void fun(int p[][4]){ } is the correct way to write the function fun(). while the others are considered only the function fun() is called by using call by reference.  
View Answer Online Compiler Report Discuss in Forum  
2.   
Which of the following statements mentioning the name of the array begins DOES NOT yield the base address?  
1: When array name is used with the sizeof operator. 2: When array name is operand of the & operator. 3: When array name is passed to scanf() function. 4: When array name is passed to printf() function. A. A B. A, B C. B D. B, D  
Answer & Explanation  
Answer: Option B  
Explanation:  
The statement 1 and 2 does not yield the base address of the array. While thescanf() and printf() yields the base address of the array.  
View Answer Online Compiler Report Discuss in Forum  
3.   
Which of the following statements are correct about the program below?  
#include<stdio.h> int main() { int size, i; scanf("%d", &size); int arr[size]; for(i=1; i<=size; i++) { scanf("%d", arr[i]); printf("%d", arr[i]); } return 0; } A. The code is erroneous since the subscript for array used in for loop is in the range 1 to size. B. The code is erroneous since the values of array are getting scanned through the loop. C. The code is erroneous since the statement declaring array is invalid. D. The code is correct and runs successfully.  
Answer & Explanation  
Answer: Option C  
Explanation:  
The statement int arr[size]; produces an error, because we cannot initialize the size of array dynamically. Constant expression is required here.  
Example: int arr[10];  
One more point is there, that is, usually declaration is not allowed after calling any function in a current block of code. In the given program the declaration int arr[10]; is placed after a function call scanf().  
View Answer Online Compiler Report Discuss in Forum  
4.   
  
Which of the following statements are correct about 6 used in the program?  
int num[6];  
num[6]=21; A. In the first statement 6 specifies a particular element, whereas in the second statement it specifies a type. B. In the first statement 6 specifies a array size, whereas in the second statement it specifies a particular element of array. C. In the first statement 6 specifies a particular element, whereas in the second statement it specifies a array size. D. In both the statement 6 specifies array size.  
Answer & Explanation  
Answer: Option B  
Explanation:  
The statement 'B' is correct, because int num[6]; specifies the size of array andnum[6]=21; designates the particular element(7th element) of the array.  
View Answer Online Compiler Report Discuss in Forum  
5.   
Which of the following statements are correct about an array?  
1: The array int num[26]; can store 26 elements. 2: The expression num[1] designates the very first element in the array. 3: It is necessary to initialize the array at the time of declaration. 4: The declaration num[SIZE] is allowed if SIZE is a macro. A. 1 B. 1,4 C. 2,3 D. 2,4  
Answer & Explanation  
Answer: Option B  
Explanation:  
1. The array int num[26]; can store 26 elements. This statement is true.  
2. The expression num[1] designates the very first element in the array. This statement is false, because it designates the second element of the array.  
3. It is necessary to initialize the array at the time of declaration. This statement is false.  
4. The declaration num[SIZE] is allowed if SIZE is a macro. This statement is true, because the MACRO just replaces the symbol SIZE with given value.  
Hence the statements '1' and '4' are correct statements.