**Answer to the Question no: 1**

Answer:

int n = 10;

int a[n], b[n];

for (int i=0; i<n; i++) {

//Write Code Here

**//scanf("%d %d",a+i, b+i);**

**int c[n];**

**c[i]=a[i];**

**a[i]=b[i];**

**b[i]=c[i];**

}

**Full code :**

| #include<stdio.h> int main() {  int n = 10;  int a[n],b[n];  for(int i=0; i<n;i++)  {  scanf("%d",a+i);  }  for(int i=0; i<n;i++)  {  scanf("%d",b+i);  }  for(int i=0; i<n;i++)  {  int c[n];  c[i]=a[i];  a[i]=b[i];  b[i]=c[i];    }  printf("array A value after swapping\n");  for(int i=0; i<n;i++)  {  printf("%d ",a[i]);  }  printf("\narray B value after swapping\n");  for(int i=0; i<n;i++)  {  printf("%d ",b[i]);   }  return 0; } |
| --- |

**Answer to the Question no: 2**

Answer:

char s[] = “banana”;

The String **S** needs **7** **bytes** in memory. Because of this is **char type** data, and char type data need **1 byte** memory. So for **“banana”** we need **6** **bytes** memory and in the string there is a hidden null value **“\0”**. So a total of **7 bytes** of memory are needed.

| #include<stdio.h> int main() {  char s [] = "banana";  int result = sizeof(s);  printf("%d",result);  return 0; } |
| --- |

**Answer to the Question no: 3**

Answer:

| #include<stdio.h> int main() {  int s [] = {1,4,7,10,15,18,5,10};   return 0; } |
| --- |

**Answer to the Question no: 4**

Answer:

Abul’s code.

char a[10];

a[0] = 'b';

a[1] = 'a';

a[2] = 'n';

a[3] = 'a';

a[4] = 'n';

a[5] = 'a';

After fixed in one line →

| char a [10] = "banana"*;* |
| --- |

Or

| char a [] = "banana"*;* |
| --- |

**Answer to the Question no: 5**

Answer:

char a[10];

gets(a);

In the upper following code, when a user put input “**1234567890**”, Then there is a **Runtime Error occour**. There is an error message, the message is *“Stack smashing detected, terminated, aborted.”* This error occurs because of buffer overflow or array overflow. In following code array size is **10**, and user input size (with a ‘\0’ null char) is **11**;

We can fix this problem using the **fgets()** function. **fgets()** is a file function.

fgets() syntax is: fgets(Name of String(char \*str), MaxNumberofCharwithNullChar(int n), File(stdin) )

After fix upper following code:

| char a[10]; fgets(a,10,stdin); |
| --- |

**[N.B]:**Max of our machine doesn’t show this type error, this error occurs Linux Os and some online compiler or online Judge contest. But we should use this solution for security purposes.

**Answer to the Question no: 6**

Answer:

An **int type array with 100 elements** takes **400** **bytes** in memory. **int** type data needs **4 bytes** memory. So for **100 elements** we need **400** **bytes** memory and in the array.

| #include<stdio.h> int main() {  int s [100];  int result = sizeof(s);  printf("%d",result);  return 0; } |
| --- |

**Answer to the Question no: 7**

Answer:

Lexicographical comparison / order also known as the “Dictionary order” or “alphabetical order”. Let we have four strings like:

1. apply
2. apple
3. boy
4. cat

In lexicographical order, **apple** comes before **apply** because the 5th character “e” of apple comes before “y” of apply.

So upper strings should in lexicographical order is

1. apple
2. apply
3. boy
4. cat

We can order strings in lexicographic order in C using the **strcmp()** function.

strcmp() returns int type value. If the value is 0, then both strings are equal, if the value is >0 then the 2nd string comes before the first string, and the value is <0 then the 1st string comes before the 2nd string.

| #include <stdio.h> #include <string.h>  int main() {  char str1[] = "abcd", str2[] = "abdc", str3[] = "abbd";  int result;   *// comparing strings str1 and str2*  result = strcmp(str1, str2);  printf("strcmp(str1, str2) = %d\n", result);   *// comparing strings str1 and str3*  result = strcmp(str1, str3);  printf("strcmp(str1, str3) = %d\n", result);   return 0; } |
| --- |

**Answer to the Question no: 8**

Answer:

A Null string in c has no values. It’s an empty Char array, one hasn't been assigned any elements.

Declare null string :

| Char str[100]; |
| --- |

In the C programming language `\0` is known as a null character.