

Homework8 Solutions

Ans.1)

Col 50 Row 50

Col 50 Row 70

Col 25 Row 50

Ans.2)

```
#include<simplecpp>
struct Pixels
{
    int color, style;
};
void showPoint(Pixels P)
{
    cout << P.color << P.style << endl;
}
main_program
{
    Pixels Point1 = {5,3};
    showPoint(Point1);
    Pixels Point2 = Point1;
    Point1.color += 2;
    showPoint(Point2);
}
```

Ans.3)

a=10 b=3

a=10 b=4 c=1

Ans.4) It will first output the sum of digits of the user input number “num”. Then it will print the sum of digits of the sum previously obtained and so on.

Ans.5) The Structure definition is within the function marksUpdate hence can't be declared in the main program. It should be outside the function to be accessible everywhere.

Ans.6): cs101_students[4] will give us the last element in array. Firstname[4] is again the fifth entry in Firstname[] array(i.e Dhantu) which is 't'. All are also similar to this.

- a. t
- b. Tanmoy
- c. n

Ans.7):

1) $\text{sqrt}(\text{pow}(p.x-q.x,2)+\text{pow}(p.y-q.y,2))$; formula for Euclidean distance
$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} .$$

2) d.center.x,d.center.y,d.radius (Recall circle from chapt5)

Ans.8):

- 1) $(n \% d != 0) \&\& (d < r)$
- 2) 1(in case of prime number)

Ans.9)

11:20
12:30
13:45

Ans.10)

10x15x6
11x16x6
10x16x11

Ans.11)

- a)f2.d
- b)f1.d
- c)f1.n
- d)frac

Ans.12):- (A) (C)

Explanation:-

Option 1: The statement 'student s[10] = {1,"Goyal",3,'A'};' declares a structure array of 10 elements but initialises only the 0th element i.e. s[0] with {1,"Goyal",3,'A'} The rest of the structure array remains uninitialised. Hence, option 1 is incorrect.

Option 2: If OP is replaced with s[counter], the statement would be s[counter+1] = s[counter]. This means that when counter is 0, the structure array s[0] will be copied to structure array s[1]. In the next iteration, when counter is 1, structure array with index 1 will be copied to structure array at index 2. This will repeated till the index reaches 9. Hence, the loop iteratively assigns the value {1,"Goyal",3,'A'} from s[1] to s[9]. Thus, option 2 is correct.

Option 3: The last cout statement prints the values at index 9. In the 2nd for loop, the index is represented as variable 'counter'. So, when the variable 'counter' has value 9, the member variables at index/counter 9 will have values (rollNumber = counter, i.e. 9; marksPhysics = counter + 10, i.e. (9 + 10 = 19); marksChemistry = counter + 20, i.e. (9 + 20 = 29); and marksMaths = counter + 30, i.e. (9 + 30 = 39). Hence, the output displayed by the cout statement would be 9, 19, 29, 39. Hence, option 3 is correct.

Option 4: None of these: This option is not applicable as there two other options which are correct.

Ans.13):-

Both are correct

Explanation:-

Implementation 1 - Correct

First the entire list is traversed to find out the no. of elements in the list. In the second while loop, we traverse the list (n-k) times from the start to get the k-th element from the last.

For example -for a list a->b->c->d->e

size = 5 and if k is given as 3

The second while loop increments the start pointer $(5-3)=2$ times and returns the value 'c'

Implementation 2 - Correct

Here we take one more pointer kth_ptr here to achieve the same functionality in a single loop. The start pointer starts traversing the list same as the previous implementation. When the start pointer reaches the kth node of the list then we start kth_ptr from the beginning. It is clear that start and kth_ptr will maintain a difference of k nodes between them. When the start pointer reaches the end of the list, the kth_ptr points to the k-th last element of the list.

For example -

a->b->c->d->e

When start pointer reaches at 'c', the kth_ptr starts incrementing. When start reaches 'e', kth_ptr points to 'c' which is the 3rd last element of the list.

Ans.14):-

0 1 2 3 7 8 6 4 5 9

Explanation:- The above program sorts the array A in increasing order. Initially, i starts with value 0 which is the index of the first element in A. This element is compared with all other elements to find the minimum. Once the minimum is found, it is swapped with the zeroth element of array A. The process continues for every value of i till n-2.

Initial contents of A : 9 1 8 2 7 3 6 4 5 0.

First swap: 0 1 8 2 7 3 6 4 5 9 (9 is swapped with 0)

Second swap: 0 1 8 2 7 3 6 4 5 9 (no change in contents, as 1 is swapped with itself)

Third swap: 0 1 2 8 7 3 6 4 5 9 (2 is swapped with 8)

Fourth swap: 0 1 2 3 7 8 6 4 5 9 (3 is swapped with 8).

Ans.15):- (C)

Explanation:- The C++ code fragment is an example of selection sort. Note that after the "for" loop terminates, the values of birthDays[index] and birthDays[n-1] are swapped, and subsequently birthDays[n-1] is not changed at all (in fact, "n" is decremented, and all subsequent operations are now done on the part of the birthDays array excluding its last element. Since we want the latest birthday (ignoring the year) to be the last element in the sorted array, we want "index" to keep track of the index of the latest birthday (excluding the year) seen so far. Since the value of index is updated to "i" whenever "compare(birthDays[i], birthDays[index])" returns true, we want "compare(date1, date2)" to return true whenever "date1" is later than "date2" (ignoring the year).

Since our dates are in the format YYYYMMDD, it is easy to see that "date1" is later than "date2" ignoring years if MMDD of "date1" is a larger integer than MMDD of "date2".

Using (date % 10000) directly gives us MMDD of a date given as YYYYMMDD, so the correct choice for "compare" is "bool compare(int date1, int date2) {return ((date1 % 10000) >= (date2 % 10000));}"

Ans.16) B

Ans.17) B

Ans.18) A,D

Ans.19):

It computes all possible permutation of a given string with duplicates allowed in the given string.

Given a string of length n it will print all the n! strings.

Ans.20):

x, arr, c_sum+num, id+1