

CLASS: XI	<b>INDIAN SCHOOL MUSCAT SECOND PERIODIC TEST</b>	SUBJECT: COMPUTER SCIENCE
	<b>SET - A</b>	
QP.NO.	VALUE POINTS	SPLIT UP MARKS
1.	i) string.h      -½ mark ii) ctype.h      -½ mark	1
2.	<b>Formal parameters</b> – parameters appear in the parameter list of function header. -½ mark  <b>Actual parameters</b> - parameters appear in the function call. -½ mark	1
3.	<b>struct address</b> - 1 mark {int houseno; char area[26]; char city[26]; }; <b>struct EMPREC</b> - 1 mark { int empno; char name[20]; address adr; float salary; };	2
4.	<b>Call by value method</b> -½ mark 1. In this method, the values of actual parameters are copied into the formal parameters and the fn. Works with its own copy of values 2. The changes in the formal parameters will not be reflected into the actual parameters. 3. The main benefit of this method is that you cannot alter the variables that are used to call the fn. And the original copy of the argument value remains intact. <b>Any example</b> -½ mark <b>Call by reference method</b> -½ mark 1. In this method, the reference to the original values is passed in the fn. 2. The same value can now be accessed by the original variable name and the reference variable name. 3. Thus, the called fn. Does not create its own copy of the original values, rather it refers to the original values with different names. 4. Thus any change with the data in the fn. Is reflected back in main(). <b>Any example</b> -½ mark	2
5.	Both #define and const define constants, however #define can only define simple constants, however #define can only define simple constants, while const can define almost any type of C constant, including things like structure classes. #define MAX 10 // define a value using preprocessor	2

	const int MAX = 10 //Define C constant integer <b>- 1 mark each</b>	
6.	Output: 40200 40500 <b>- 1 mark for each correct line</b>	2
7.	for declaration & input statements <b>-½ mark</b> for correct logic <b>-2 marks</b> for displaying <b>- ½ mark</b>	3
8.	for declaration & input statements <b>-½ mark</b> for correct logic <b>-2 marks</b> for displaying <b>- ½ mark</b>	3
9.	for declaration & input statements <b>-½ mark</b> for correct logic <b>-3 marks</b> for displaying <b>- ½ mark</b>	4