Indian Institute of Technology Mandi IC150: Quiz 2 I5th May 2013, 8:00-8:50 a.m. Answer all questions. No calculators or cellphones. Ansimum marks: 25 O) The minimum attendance requirement for a teacher in his/her own lectures should be: [½] a) 0% b) 50% c) 75% d) 100% IFill in the blanks: [4½] a) An example of Fibonacci numbers in Nature is b) The array a is declared by char a[10][10] and it starts at memory location 100. The address of element a[5][3] is c) A pointer p that may be used to point to any data type must be declared by	Name					- 1 -			Roll N	lo:
Answer all questions. No calculators or cellphones. Maximum marks: 25 (a) The minimum attendance requirement for a teacher in his/her own lectures should be: [½] (a) 0% (b) 50% (c) 75% (d) 100% (e) 75% (d) 100% (f) Fill in the blanks: (a) An example of Fibonacci numbers in Nature is (e) The array a is declared by char a[10][10] and it starts at memory location 100. The address of element a[5][3] is (f) A pointer p that may be used to point to any data type must be declared by (g) Given the declarations: (g) Struct point { int x, y; } p1; (g) Struct point *p2 = &p1 (g) Two ways of accessing the value of the x component of p1 are (g) and that of insertion sort is (g) and that of insertion sort is (g) Complete the C code: (g) Float Length; Length x, y; (g) Complete the C code: (g) Float Length; Length x, y; (g) Write a C function char *GenString(char ch, int len). This function allocates storage for a string of len characters each having value ch. It returns the new string. (g) Eg. GenString('a', 3) returns the string "aaa" and GenString('z', 0) returns the empty string "". (har *GenString(char ch, int len) (har *GenString(char ch, int len)) 1	2	3	4	5	6	7	Total	Indian Institu	nte of Technology Mandi
a) 0% b) 50% c) 75% d) 100% f) Fill in the blanks: [4½] a) An example of Fibonacci numbers in Nature is b) The array a is declared by char a[10][10] and it starts at memory location 100. The address of element a[5][3] is c) A pointer p that may be used to point to any data type must be declared by *p. d) Given the declarations: struct point { int x, y; } p1; struct point *p2 = &p1 Two ways of accessing the value of the x component of p1 are and e) The best-case time complexity of selection sort is and that of insertion sort is f) The C string "count me in" requires at least bytes of memory. g) Complete the C code: float Length; Length x, y; 2) Write a C function char *GenString(char ch, int len). This function allocates storage for a string of len characters each having value ch. It returns the new string. Eg. GenString('a', 3) returns the string "aaa" and GenString('z', 0) returns the empty string "". [3] char *GenString(char ch, int len) {										
a) 0% b) 50% c) 75% d) 100% l) Fill in the blanks: [4½] a) An example of Fibonacci numbers in Nature is b) The array a is declared by char a[10][10] and it starts at memory location 100. The address of element a[5][3] is c) A pointer p that may be used to point to any data type must be declared by*p. d) Given the declarations: struct point { int x, y; } p1; struct point *p2 = &p1 Two ways of accessing the value of the x component of p1 are and e) The best-case time complexity of selection sort is and that of insertion sort is f) The C string "count me in" requires at least bytes of memory. g) Complete the C code: float Length; Length x, y; 2) Write a C function char *GenString(char ch, int len). This function allocates storage for a string of len characters each having value ch. It returns the new string. Fg. GenString('a', 3) returns the string "aaa" and GenString('z', 0) returns the cmpty string "". [3] char *GenString(char ch, int len) {	Answ	er all que	stions.	No calci	ulators	or cellpl	nones.	•	Maximum ma	arks: 25
b) 50% c) 75% d) 100% 1) Fill in the blanks: [4½] a) An example of Fibonacci numbers in Nature is			um atten	dance re	equirem	ent for a	teacher	in his/her ov	wn lectures shou	ld be: [½]
c) 75% d) 100% 1) Fill in the blanks: [4½] a) An example of Fibonacci numbers in Nature is										
d) 100% 1) Fill in the blanks: [4½] a) An example of Fibonacci numbers in Nature is	/									
a) An example of Fibonacci numbers in Nature is	/									
b) The array a is declared by char a[10][10] and it starts at memory location 100. The address of element a[5][3] is	1) Fil	in the bl	lanks:							$[4\frac{1}{2}]$
b) The array a is declared by char a[10][10] and it starts at memory location 100. The address of element a[5][3] is	a)	An evai	mnle of l	Fibonac	ci numb	erc in N	ature is			
The address of element a [5][3] is c) A pointer p that may be used to point to any data type must be declared by*p. d) Given the declarations: struct point { int x, y; } p1; struct point *p2 = &p1 Two ways of accessing the value of the x component of p1 are and e) The best-case time complexity of selection sort is and that of insertion sort is f) The C string "count me in" requires at least bytes of memory. g) Complete the C code: float Length; Length x, y; 2) Write a C function char *GenString(char ch, int len). This function allocates storage for a string of len characters each having value ch. It returns the new string. Eg. GenString('a', 3) returns the string "aaa" and GenString('z', 0) returns the empty string "". [3] char *GenString(char ch, int len) {			•							
c) A pointer p that may be used to point to any data type must be declared by *p. d) Given the declarations: struct point { int x, y; } p1; struct point *p2 = &p1 Two ways of accessing the value of the x component of p1 are and e) The best-case time complexity of selection sort is and that of insertion sort is f) The C string "count me in" requires at least bytes of memory. g) Complete the C code: float Length; Length x, y; 2) Write a C function char *GenString(char ch, int len). This function allocates storage for a string of len characters each having value ch. It returns the new string. Eg. GenString('a', 3) returns the string "aaa" and GenString('z', 0) returns the empty string "". [3] char *GenString(char ch, int len)	U)		•		-				-	ation 100.
d) Given the declarations: struct point { int x, y; } p1; struct point *p2 = &p1 Two ways of accessing the value of the x component of p1 are and		THE auc	11088 01 (CICIIICIII	a[J][2] 18			·	
d) Given the declarations: struct point { int x, y; } p1; struct point *p2 = &p1 Two ways of accessing the value of the x component of p1 are and	c)	A point	er p that	may be	used to	point to	any da	ta type must	be declared by	
struct point { int x, y; } p1; struct point *p2 = &p1 Two ways of accessing the value of the x component of p1 are and					* p.					
e) The best-case time complexity of selection sort is and that of insertion sort is f) The C string "count me in" requires at least bytes of memory. g) Complete the C code: float Length; Length x, y; 2) Write a C function char *GenString(char ch, int len). This function allocates storage for a string of len characters each having value ch. It returns the new string. Eg. GenString('a', 3) returns the string "aaa" and GenString('z', 0) returns the empty string "". [3] char *GenString(char ch, int len) {	d)	s Two wa	truct truct ays of a	point point ccessing	*p2 the val	= &p1	;		o1 are	
f) The C string "count me in" requires at least bytes of memory. g) Complete the C code: float Length; Length x, y; 2) Write a C function char *GenString(char ch, int len). This function allocates storage for a string of len characters each having value ch. It returns the new string. Eg. GenString('a', 3) returns the string "aaa" and GenString('z', 0) returns the empty string "". [3] char *GenString(char ch, int len) {	e)					of selec	tion sort	t is		
memory. g) Complete the C code:float Length; Length x, y; 2) Write a C function char *GenString(char ch, int len). This function allocates storage for a string of len characters each having value ch. It returns the new string. Eg. GenString('a', 3) returns the string "aaa" and GenString('z', 0) returns the empty string "". [3] char *GenString(char ch, int len) {		and tha	t of inser	rtion sor	t is	_	_			
2) Write a C function char *GenString(char ch, int len). This function allocates storage for a string of len characters each having value ch. It returns the new string. Eg. GenString('a', 3) returns the string "aaa" and GenString('z', 0) returns the empty string "". [3] char *GenString(char ch, int len) [4]	f)		_	ount	me in	" requir	es at lea	st		bytes of
ates storage for a string of len characters each having value ch. It returns the new string. Eg. GenString('a', 3) returns the string "aaa" and GenString('z', 0) returns the empty string "". [3] char *GenString(char ch, int len) {	g)	Comple	ete the C	code: _		_	f	loat Leng	gth; Length	x, y;
	ate Eg. turn ch {	s storage GenSt: is the em ar *Ge	for a str ring(' apty strin nStrir	ing of 1 a', 3 ig "". ng(cha	en char) return r ch,	int	ach hav ring "a len)	ing value ch	. It returns the n	new string. , 0) re-

}

[5]

3) Answer briefly:

(a) What is meant by the Golden Ratio?

(b) Name one algorithm that uses the divide-and-conquer strategy.

(c) A character array is used to store the name of any State of India. What size of array would you choose? Justify your answer.

(d) What is the value of n after the following code is executed? Explain.

```
#define mystery(a, b) a * b
n = mystery(2+2, 3+3);
```

- (e) Given the declaration struct s { int a, b, c;} s1, s2; which one of the following is **not** valid? Explain.
- a) s1 = s2;
- b) s1.a = s2.b + s2.c;
- c) if (s1 > s2) s1.a++;
- d) if (s1.a < s2.a) s1.a++;

Name: - 3 - Roll No:

4) Consider the C code below.

[4]

- (a) What is printed out by this program?
- (b) What is the purpose of this function?
- (c) Will the function achieve this purpose for all values of strings a and b? Justify your answer.

```
#include <stdio.h>
void Doit(char *s, char *t)
{
    while (*t) t++;
    t--;
    while (*s)
      {
        *t = *s;
        t--; s++;
    }
}

void main()
{
    char a[] = "one";
    char b[] = "two";
    Doit(a, b);
    puts(a);
    puts(b);
}
```

5) It is desired to sort an array in descending order using (a) selection sort and (b) insertion sort. In the figures below, show the contents of the array after the first and second iterations. In each iteration, circle the elements that you modify. [2]

(a) Selection Sort

Initial	23	20	7	-2	90	10	5	50
Iteration 1								
Iteration 2								

(b) Insertion Sort

Initial	23	20	7	-2	90	10	5	50
Iteration 1								
Iteration 2								

6) A C array is declared by int m[A][B][C]. Its starting address in memory is mbase. Derive an expression for the address of m[i][j][k]. [2]

7) The following structures were developed to represent a book.

```
struct book
{
  struct page frontCover;
  struct page backCover;
  struct binding bindingType;
  struct page initial[10];
  struct page mainPages[1000];
};
```

```
struct page
{
   char pageText[1000];
   int pageNumber;
   int fontType;
};

struct binding
/* 1 Soft, 2 Hard, 3 Cloth */
{
   int type;
}
```

[4]

The following are statements about the above representation. State if each is true or false and explain your answer.

- a. The number of characters allotted for the front is too much for an average front cover.
- b. A book can contain over 10,000 main pages.
- c. The same page number can be assigned to different pages.
- d. Each character in the book can be allotted a different font.