E1 Questions
Topics in L01 through L08

Select a different topic than those already addressed or create a substantially different question about the same topic.		ANSWER	COMMENTS
#. Choose a Lecture Topic Add your name	Add your multiple choice question text, images, and choice(s) here.	Your answer to your question or leave it blank for others to answer.	Provide feedback for a question without any feedback yet. Include your name
0. Linux file copy by Debra Deppeler	Which command is correct for copying all **.txt** files from your *handin* directory to your *current working directory*? Assume that your handin directory is /p/course/cs354/handin/p1/badger/ A. copy /p/course/cs354/handin/p1/badger/*.txt . B. cp /p/course/cs354/handin/p1/badger/*.txt . C. cp . /p/course/cs354/handin/p1/badger/*.txt D. copy . /p/course/cs354/handin/p1/badger/*.txt E. mv /p/course/cs354/handin/p1/badger/*.txt cwd	B	
1.Java vs C **by Brandon Kenter**	Which of the following statements is false? A. C is a compiled language, not interpreted B. Java is considered to be a higher level language than C C. C is a procedural language D. C code can be compiled on one machine and then shared across multiple machines	D	
2. Coding in C remotely **By Michelle Tan**	Which of the following lines are incorrect when coding in C remotely to create a C file called "test.c" in the path "username/private/cs354/"? ssh username@best-linux.cs.wisc.edu //1 *type password*	A	This is a good, simple question, but it still made me wonder whether "cd public" was valid code I had missed.

E1 Questions
Topics in L01 through L08

	cd public //2 cd cs354 //3 mkdir test.c //4 vim test.c //5 A. 2 and 4 B. 1 and 5 C. 2 and 5 D. 4 and 5 E. 4 only		- Brandon Kuhl
3.Build an executable **By Alex Huang**	Which of the following commands will compile a C source file into an **executable file** for the purposes of CS354? A. gcc -S <source-file-name> -Wall -m32 -std=gnu99 <executable-name> B. gcc <source-file-name> -Wall -m32 -std=gnu99 -o <executable-name> C. gcc -E <source-file-name> -Wall -m32 -std=gnu99 -o <executable-name> D. gcc <source-file-name> -Wall -m64 -std=gnu99 -o <executable-name>	B	This is a good question because it can always be confusing to figure out which command should be used to create the proper executable file. -Aalyaan Feroz
4.C Program Structure **By Brandon Kuhl**	Which of the following is true of the phrase “return-by-value”? I. It refers to a means of a function to share data II. It refers to passing data into a function’s parameters III. It refers to making a copy of the return value to replace the function in the caller IV. The function “passes out” a value A. I. B. II. C. III. D. I, III E. I, III, IV	E	

E1 Questions
Topics in L01 through L08

5. C Control Flow
 	
By Harry Zhao
	Which of the following will be printed?
	int main(){
	if (8)
	  if(-1)
	  printf("A");
	else
	  printf("B");
	  printf("C");
	}
	A. A
	B. B
	C. BC
	D. AC
	E. ABC
		D	This is a good question as it lacks indentations and parentheses in some places, testing the users knowledge on the functioning of if else conditions in C. It also tests the understanding of numbers having a true/ false value in C.
			-Anvit Thekkatte

6. C Pointers
 	
Cathy Cao
	What is the value of i after the following code is executed?
	
	int main() {
	  int i = 100;
	  int *ptr = NULL;
	  int *ptr2 = NULL;
	
	  ptr = i;
	  ptr2 = &i;
	  ptr = 200;
	  *ptr2 = 300;
	
	  return 0;
	}
		i = 300
		
		ptr is assigned the initial value of i, which is 100. ptr2 is assigned the address of i. Then, ptr is assigned to the value 200. Finally, ptr2 is dereferenced, assigning the value of what
			When you mention that ptr is assigned the initial value of i, and ptr is assigned to the value of 200 is this an integer representation of ptr's memory location?
			-Caleb Engel

E1 Questions
Topics in L01 through L08

		it is pointing at (which is i) to 300.	

7. 1D Arrays (SAA)
 	
Caleb Engel
	int main() {
	  int my_array[4] = {1,2,3,4};
	  int a = *my_array; //a
	  int b = *(my_array+0); //b
	  int c = *my_array[0]; //c
	  int d = *(&my_array[0]); //d
	  int e = my_array[0]; //e
	  int f = &my_array[0]; //f
	
	  return 0;
	}
	
	What variables will contain an int value of 1?
		a, b, d,e
		
		c attempts to store an int* to an int
		
		f will store the int value associated with a[0]'s memory address
			This is a good question because it shows both indexing and address arithmetic while showing pointer and array functionality that may be used in other scenarios but does not have the desired results.
			-Autumn Trzebiatowski

8. Arrays and Pointers
 	
Tavish Vats
	int main()
	{
	 int arr[] = {20, 30, 40};
	 printf(%d, *(arr + 1));
	}
	
	What is the output of the code above?
	A. 40
	B. 30
	C. 20
	D. Compiler Error
		30	This is a good question, as it combines Standard I/O and array arithmetic. It also makes the test taker wonder if printf will work with that sort of indexing.
			
			-Ryan Boukhankov

E1 Questions
Topics in L01 through L08

9.Passing Arrays
 	
Soyoon Lee
	Which of the following is the correct way to pass the array to a function?
	void print_average(int (A), int size)
	{
	  int sum = 0;
	  for (int i = 0; i < size; i++)
	  {
	  sum += arr[i];
	  }
	  double avg = sum / size;
	  printf("Average is: %f\n", avg);
	}
	
	int main()
	{
	  int arr[] = {354, 9, 29, 2020, 621};
	  int length = sizeof(arr) / sizeof(arr[0]);
	  print_average((B), length);
	  return 0;
	}
	
	(A) (B)
	1 - *arr *arr
	2 - *arr arr
	3 - arr[] *arr
	4 - &arr arr[]
		2	While this is a very simple question in concept, this is a mistake that a lot of programmers make a lot and it helped me make sure when to use the star to denote the ptr.
			-Shashank Bala

E1 Questions
Topics in L01 through L08

10. Arrays on Heap **Archer Li**	Which is the correct lines of code we need to free a heap allocated 2d array A from memory: A: free(A); B: for (int i = 0; i < A.length; ++i){free(A[i]);} C: A = NULL; D: for (int i = 0; i < A.length; ++i){free(A[i]);} free(A); E: for (int i = 0; i < A.length; ++i){A[i]=NULL);} A=NULL;	D	D is the correct line of code. b/c we not only need to free() A itself. We also need to free() the pointers that A stored. Although by doing only free() without set all pointer =NULL would create dangling pointers. It still freed memory beforehand as the question asked. (By Archer Li)
11. C strings **Rixi Lu**	Which of the following ends without null char '\0': a. char str1[] = {'M', 'I', 'D', 'T', 'E', 'R', 'M'}; b. char str2[] = "MIDTERM"; c. char *str3 = "midterm"; d. char str4[8] = "Midterm";	a	This is a good question–it addresses the subtle and important difference between what is regarded as a 'string' and what is regarded as a 'char array' in C, in terms of having or not having a null character. Cathy Cao
12.string.h functions **By Anvit Thekkatte**	char *strp = "hello"; char str1[7] = "hello"; char str2[20] = "world"; //Line A Which of the following options for Line A execute without an error: A. strcpy(strp, str1); B. strcat(str1,str2);	C	This is a great question because it can be difficult to remember which types of strings can or cannot be changed after declaration. Saurav Mathur

E1 Questions
Topics in L01 through L08

	C. strcat(str2,str1); D. strcpy(strp, str2);		
13. 2D Arrays on Heap **By Milica Andric**	Which of the following are equivalent to m[i][j]? a.) *(m[i] + j) b.) (*(m + i))[j] c.) *(*(m[i] + j)) d.) *(*(m + i) + j)		This is a challenging question. It addresses how to access elements in a 2 dimensional array using the concept of dereferencing and indexing. Tavish Vats
14. 2D Arrays on Stack **By Autumn Trzebiatowski**	Given: int array[2][5]={{.....},{.....}}; What is equivalent to array[1][4]? a.) *(*(array+5)+2) b.) *(*array+4)+1); c.) *(*array +2*4+1) d.) *(array+8)	d	I believe you meant c to be the correct answer.

15. Arrays: Stack vs Heap
By Aalyaan Feroz
	Given int **a, what is the correct way to completely initialize the array if the array has 2 rows and 2 columns
	a) int **a = malloc(sizeof(int) * 2 * 2);
	b) int **a = malloc(sizeof(int) * 2);
	a[0] = malloc(sizeof(int) * 2);
	a[1] = malloc(sizeof(int) * 2);
	c) int **a = malloc(sizeof(int) *2);
	a[0] = malloc(sizeof(int) * 2 * 2);
	a[1] = a[0];
	d) int **a = malloc(sizeof(int) * 2);
		B	This is a great question because the answers are similar and it really makes you need to understand the concept to answer it. - Joseph Knuth

E1 Questions
Topics in L01 through L08

16. Array Caveats **By Nicholas Spevacek**	Which of the following is statements is false? a.) Arrays have no bounds checking b.) Arrays cannot be return types c.) An array argument must match its parameter's type d.) Stack allocated arrays need their first dimension to be specified	d	This is a good question since it helps identify differences between Arrays in C and Java, which can sometimes be confusing. It also highlights the requirements needed for allocating Arrays on the stack. Annabelle Zhang
17. CLAs and PAs **By Saurav Mathur**	Consider the following command $gcc helloworld.c -m32 -std=gnu99 helloworld 1) The number of program arguments are 5. 2) The number of program arguments are 4. 3) The total argument count (command line arguments) is 5. 4) The program name is gcc. Which of the following are correct a.) 1 and 3 only b.) 2 and 3 only c.) 1, 3 and 4 only d.) 2, 3 and 4 only	d	
18. Structs **By Joseph Knuth**	Which of the following best describes what structs can contain: 1) Data members and functions 2) Functions only 3) Data members only 4) Strings only		

E1 Questions
Topics in L01 through L08

19. Passing Structs w/o ptr			

20. Pointers to Structs
 	
By Annabelle Zhang
	The following code assigns a student A's grade in cs354. Please choose the correct code that can fill in the blank for the code to work respectively.
	
	typedef struct{
	Char coursename[42];
	Int grade;
	} Course;
	
	typedef struct{
	Char name[42];
	Course compsci;
	}Student;
	
	int main(){
	Student *ptr = _____(a)_____;
	_____(b)_____; // assign name to the student
	_____(c)_____; // assign name to the student
	_____(d)_____; // assign grade to the student
	}
	
	A. malloc(sizeof(Student)), strcpy("your name", ptr->name), strcpy("cs354", ptr->ptr.compsci.coursename), (*ptr).compsci.grade = 100
	
	B. malloc(sizeof(Student) + sizeof(Course)), ptr->name = "your name", ptr->compsci.coursename = "cs354", ptr->compsci->grade = 100
	
	C. malloc(sizeof(Student)), strcpy(ptr->name, "your name"), strcpy(ptr->compsci.coursename, "cs354"), ptr->compsci.grade = 100
		C	

E1 Questions
Topics in L01 through L08

	D. malloc(sizeof(Student)), *ptr.name = “your name”, *ptr.compsci->coursename = “cs354”, *ptr->compsci.grade = 100		

21. Standard I/O
Ryan Boukhankov
	Which of the following will be output by the following lines of code? (‘a’ is 97 in ASCII)
	int a = 101;
	int b = 102;
	printf(“Number: %i, Letter: %c\n”, a, b);
	
	A)Number: 101, Letter: 102
	B)Number 101, Letter e
	Number 102, Letter f
	C)Number: 101, Letter f
	D)The code cannot run, a and b are strictly int values.
		C	This is a really good question that helps us understand the conversion between ASCII char to int.
			Nirmit Jallawar

22. String I/O
 	
Luis Cazarin Quiroga
	What will be the output of the following code?
	#include<stdio.h>
	
	int main() {
	
	char buffer[26];
	int v1 = 30 % 7;
	int v2 = 7;
	
	sprintf(buffer, "Product of %d and %d is %d", v1, v2, v1 * v2);
	
	printf("%s", buffer);
	
	return 0;
		a	

E1 Questions
Topics in L01 through L08

	} a) Product of 2 and 7 is 14 b) Segmentation fault c) Product of 1 and 7 is 7 d) Product of 7 and 2 is 14		
23. File I/O **Aabha Mishra**	If a FILE pointer is NULL, in the following example, what does it mean.? FILE *fp; fp=fopen("example.txt","w"); A) Unable to open a file named example.txt B) example.txt is not available on disk C) Hard disk has hardware problems. D) All the above	D	Great to look up all of the potential reasons for a NULL file pointer value. The wording on this question could be modified for better clarity. If fp is NULL, in the following example, which of the following options could have caused this? Caleb Engel

23. File I/O
Nirmit Jallawar
	Which of the following is the best way to handle closing a file? (Assume the FILE pointer to be fptr)
	
	A) fptr = NULL;
	B) free(fptr);
	C) if(fclose(fptr) != 0){
	//handle error here
	}
	D) None of the above
		C	

E1 Questions
Topics in L01 through L08

24. Copying text files			
25. Virtual Address Space (IA-32/Linux) Shashank Bala	Which of the following segments of C's abstract memory model stores literal strings? A) Code Segment B) Data Segment C) Heap D) Stack	A	
26. Three Faces of Memory			
27. Linux Processes and Address Spaces			
28. p1 encode/decode			
29. p2A n_in_a_row			
30. p2B magic square			

E1 Questions
Topics in L01 through L08

31.			
32.			
33.			
34.			
35.			