CS 211

SUMMER 2020

Multiple Choice:

1. A heap in terms of memory management in a computer system, is:
   1. a region of main memory where dynamic memory is allocated
2. A stack:
   1. grows towards lower addresses
3. An activation record:
   1. is present per function invocation
4. A Von Neumann Machine:
   1. maintains both instructions and data in main memory
5. The heap:
   1. can be managed with calls to malloc and free
6. Pointers are:
   1. Memory addresses of variable and should be declared to match the data type it points to
7. Which of the following does not have a corresponding primitive (built-in) data type in C?
   1. Stack Pointers
8. Assume the following function foo() in C:  
     
   int foo(){  
      int a = 6;  
      int b = 7;  
      int c = 15;  
      int d = 14;  
      int z = bar(d, c, a, b);  
      return z;  
   }
   1. 7, 6, 15, 14, return address
9. Register EBP and ESP:
   1. manage the stack frame of a function call
10. The EAX register:
    1. holds the return value
11. Which of these is true about the ret instruction?
    1. It assumes that the top of the stack has the return address
12. The EIP register:
    1. holds the address of the next instruction to be executed

Fill In Blank:

1. Assume we have a 12-bit signed integer using twos complement representation. What is the **lowest** value (in decimal) this integer can represent?
   1. -212-1 = -211 = -2048
2. Assume we have an 12-bit unsigned integer. What is the**largest** value (in decimal) this integer can represent?
   1. 212 – 1 = 4095
3. Assume the following assembly code:  
     
   movl $2, %eax  
   movl $8, %ecx  
   leal (%ecx, %eax, 4), %eax  
   pushl %eax  
   movl $9, %edx  
   popl %ecx  
   addl %ecx, %edx  
   movl %edx, %eax  
     
   What are the final values (in decimal) stored in the following registers after executing the assembly instructions above?:
   1. %eax: 25, %ecx 16, %edx: 25

Open Ended

1. Complete the implementation for the following function alloc2DArray() that dynamically allocates a 2D array via malloc() with numRows rows and numCols columns, initializes all elements to zero, and returns a pointer to it.

|  |
| --- |
|  |

* 1. int \*\*alloc2DArray(int numRows, int numCols){  
      int x, y;  
      int \*\*arry

// allocate row pointers  
 arry = (int \*\*) malloc(sizeof(int\*) \* numRows);

// allocate rows  
 for(x = 0; x < numRows; x++){  
 arry[x] = (int \*) malloc(sizeof(int) \* numCols);  
 }

// zero all rows  
 for(x = 0; x < numRows; x++){  
 for(y = 0; y < numCols; y++){  
 arry[x][y] = 0;  
 }  
 }

// return pointer  
return arry;  
}

1. Assume we have the following node structure:  
     
   typedef struct node{  
       int value;  
       struct node \*next;  
   } Node;  
     
   Complete the implementation of the following function *insertLL()*. This function should insert a value into the linked list and return 0. However, if the value already exists within the linked list, the value should not be added and the function should return 1.  
     
   Node \*head;  // Global head of linked list  
     
   int insertLL(int value){

|  |
| --- |
|  |

Node\* ptr = head;  
Node \*prev = NULL;  
Node \*curr;  
  
curr = (Node \*) malloc(sizeof(Node));  
curr->value = value;  
  
if(\*head == NULL){  
 head = curr;  
 curr->next = NULL;  
}  
  
else{

//traverse linked list  
while ( ptr != NULL && ptr->value != value){  
 prev = ptr;  
 ptr = ptr.next;  
}  
  
//found duplicate  
if(ptr->value == value)  
 return 1;  
  
//reach NULL, no duplicate  
else if(ptr == NULL){  
 prev->next = curr;  
 curr->next = NULL;  
}

}

1. Assume we are using an 8-bit signed integer using twos complement representation.  
   Convert -47 to binary.

Please show all work for full credit.

|  |
| --- |
|  |

Step 1 find +47: 00101111 == 1\*(2^5) + 1\*(2^3) + 1\*(2^2) + 1\*(2^1) + 1\*(2^0)  
 32 + 8 + 4 + 2 + 1 = 47  
Step 2 ones complement: 11010000  
Step 3 Twos Complement (add 1): 11010001  
FINAL ans: 11010001 === -128 + 64 + 16 + 1 = -47

1. Assume we are working with an 8-bit floating point representation, where there is 1 sign bit, 3 exponent bits, and 4 mantissa bits.  
     
   What is the following binary floating point value in decimal?  
   10110101   
     
   Please show all work for full credit.

|  |
| --- |
|  |

1 011 0101  
  
  
S = (-1)^1 = -1  
E = exp - bias = (2^1 + 2^0) - (2^(3-1)-1) = 3 - 3 = 0  
F = 1.0101 (Normalized value bc exp not all 0 or 1)  
  
Value = S \* 2^E \* F  
= -1 \* 2^0 \* 1.0101  
= -1 \* 1.0101 (convert to decimal and add sign last)  
= 2^0 + 2^-2 + 2^-4  
= 1 + 0.25 + 0.0625  
= -1.3125