CSE2421 AU12 MIDTERMA NAME _______KEY (orange/purple)___

Instructions: This test is for your eyes only. The information on this exam is NOT to be shared in any way. This exam is 100% your own work.
NOTE: If there is not an answer in the blank provided, the answer is wrong.
Warning: If I see you trying to page through your exam to compare questions with your neighbor, I will move you. If I see you looking at your neighbor's paper, I will move you. If I suspect that you are talking with your neighbor, I will move you. The time and attention this takes away from you taking your exam is not available for makeup; that is, you get no extra time for not paying strict attention to your own paper.

YOU MUST TURN IN EVERY PAGE OF THIS EXAM including the ref card page

MATCHING (10 questions; 2 points each) Match each of the given concepts to the most appropriate definition. You can only use each definition once, but not all definitions will be used.

GIVEN CONCEPT

_________ gdb **I**

_________ break statement **L**

_________ free **N**

_________ argc (by convention) **H**

_________ continue statement **D**

_________ member **E**

_________ #include <file> **K (J)**

_________ typecast **G**

_________ preprocessor **J (K)**

_________ #define VAR value **B**

DEFINITION/MEANING

A. debugger command
B. gives a meaningful name to a constant in your program
C. used to exit the current function being executed
D. stops iteration; updates itself; begins execution from the top of the loop
E. a variable in a structure
F. element of an array
G. makes a variable of one type act like another type for a moment in time
H. the number of arguments for your main function
I. debugger environment
J. provides the ability for the inclusion of header files
K. inserts pre-written code into a current program
L. is used to quit the current iteration
M. gives an initial value to a variable in your program
N. deallocates memory
O. specifies that the value of a declared variable is no longer accessible

MULTIPLE CHOICE (20 questions; 2 points each). Put your answer in the blank provided.

_______ What is the correct value to return to the operating system upon the successful completion of a program?
A. -1 B. 1 C. 0 D. Programs do not return a value.
Void vs int return type

_______In order to properly use a variable… **NOT GRADED**
A. the variable must have a valid type
B. the variable name can not be a keyword (part of the C syntax)
C. the variable name must begin with a letter **or underscore**
D. all of the above

_______Which of the following gives the value stored at the address pointed to by pointer a?
A. a; B. val(a); C. *a; D. &a;

_______Function prototypes are use:

A. because they tell the compiler that a function is declared later
B. because they make the program more readable
C. because they allow the programmer to see a quick list of functions in the program along with the arguments of each function
D. all of the above

_______Adding to a pointer that points to an array will:

A. cause an error
B. increase the value of the element that the point is pointing to
C. cause the pointer to point to the next element in the array
D. none of the above

_______Because of variable scope:

A. variables created in a function cannot be used in another function
B. variables created in a function can be used in another function
C. variables created in a function can only be used in the main function
D. none of the above

_______Which of the following is a logical true value?

A. 1 B. 66 C. .1 D. -1 E. All of the “above”

_______Evaluate !(1 && !(0 | | 1)).

A. True B. False C. Not able to evaluate

_______How many times is a do while loop guaranteed to execute?

A. 0 B. Infinitely C. 1 D. It depends

_______Which of the following accesses a variable in structure *b?

A. b->var; B. b.var; C. b-var; D. b>var;

_______What is the result of the following code?

```
int x=0;
switch(x)
{
  case 1: printf( "One" );
  case 0: printf( "Zero" );
  case 2: printf( "Hello World" );
}
```

A. One B. Zero C. Hello World D. ZeroHello World

_______Which of the following gives the memory address of the first element in array foo, an array with 100 elements?

A. foo[0]; B. foo; C. &foo; D. foo[1];

_______What object do you use to represent a file in C?

A. FILE* B. fopen C. printf D. fprintf

_______How do you write a string of text into a file?

A. Open file and use fprintf.
B. Open a file and use printf, the output will go to the file instead of the screen.
C. Open a file, and use fputc repeatedly.
D. Use fread to read data into the file.

_______Which is a good use for typecasting?

A. To allow division of two integers to return a decimal value.
B. To allow your program to use nothing but integers.
C. To change the return type of a function.
D. To swap variables rapidly.

_______What type is argv?

A. char *
B. int
C. char **
D. It's not a variable

_______What is argv[0]?

A. The number of arguments to the program
B. The name of the program
C. The first argument to the program
D. This syntax is illegal

_______What does this code do?

```
float x;
int *ptr = (int *) (&x);
*ptr = *ptr << n
```

A. allows a float to be left shifted by n bits
B. allows an int to be left shifted by n bits
C. allows an address to be left shifted by n bits
D. don’t have enough information (n and x not initialized…)

_______Given the code below, what happens to the following loop (there is no syntax error)?

```
while (‘x’ <> “x”)  { execute statements }
```
<> should be !=

A. the statements in the block never execute
B. the statements in the block execute infinitely
C. it depends on what statements are inside the block what if break statement
D. unknown

_______Why do you sometimes need to use & on the scanf?

A. because you can’t change a value inside a function
B. the address of a value can be accessed outside a function
C. both A and B
D. None of the above

SHORT ANSWER (12 pts)

_______ Evaluate the following C arithmetic statement and put the final result in the blank provided

```
int a = 10, b = -25, c=0;
c = 1 + a - ++ b;              //35
```

What is printed by this fragment of code?

```
enum Liquid {OUNCE=1, CUP=8, PINT=16, QUART=32, GALLON=128 };
enum Liquid jar;
…
jar = QUART;              // what is the value of jar here? _____________ (32)
jar = jar + PINT;         // what is the value of jar here? _____________ (48)
```

For each of the two sections of code, fill in the blank by writing one line of code to declare a variable, *teststruct*, to be a structure of type mystruct (3 pts each) :

```
struct mystruct
{ // stuff listed here
};
          ___________________________________   (struct mystruct teststruct;)
```

```
typedef struct my_struct
{ // stuff listed here
} mystruct;
          ___________________________________  (mystruct teststruct;)
```

CODE EVALUATION #1 (8 pts)

```
char input[101], save_first_letter;
char *inptr = input;
int first_letter = 1;
 if (isalpha(*inptr))                   // the current character is alphabetic
{   *inptr = tolower(*inptr);           // convert character to lower case
     if (first_letter)                  // looking at the first letter of a word
    {    save_first_letter = *inptr;    // keep track of first letter of the current word
          first_letter = 0;             // set flag to designate have the first letter of the word already
         *inptr = *(inptr+1);           // move right one position
    }
   else                                 // alphabetic char but not the first letter
         *(inptr-1) = *inptr;           // move left one position
}
```

CODE EVALUATION #2 (12 pts) – quite a few options for the first 3 blanks

```
// deletes the specified node pointed to by 'ptr' from the list
void deletenode( struct node *ptr )
{   struct node *temp, *prev;

    temp = ptr;                        // node to be deleted

    prev = head;                       // start of the list, will cycle to node before temp

    if( temp == prev )                 // are we deleting first node

    {   head = head->next;             // moves head to next node

        if( end == temp )              // is it end, only one node?

             end = end->next;          // adjust end as well

        free( temp );                  // free space occupied by node

    }

    else                               // if not the first node, then

    {  while( prev->next != temp )     // move prev to the node before

             prev = prev->next;        // the one to be deleted

        prev->next = temp->next;       // link previous node to next

        if( end == temp )              // if this was the end node,

             end = prev;               // then reset the end pointer

        free( temp );                  // free space occupied by node

     }

}
```

CODE EVALUATION #3 (8 pts)

```
#include <stdio.h>
main()
{
  char input = 0b10010101;
  unsigned char mask = 0b10000000;
  int x;

  for (x = 1; x < 9 ; x++)
    {
      if ((input & mask) != 0)
           printf("sport = %d\n", x);
      mask = mask >> 1;
    }
  return 0;
}
```

What is the last line of output? ______________________________ sport = 8

Are parentheses needed around the "input & mask" part of the condition? __________________ Yes

What is the value of x when the loop completes? ___________ 9

What is the value of the variable "mask" at the end of the program? _________________________ 0