CS111: Computer Programming **Name:** _____
Jul-Dec 2017
End Sem.
6/12/2017
Time Limit: 180 Minutes Roll Number _____

Read the following instructions carefully before you start writing the answers

1. This exam contains 10+1 pages (including this cover page) and 8 questions. Total of points is 150. **Don't forget to write your name, roll number and sign in the place provided.**

2. **Answers to Question no.** $1, 2, 3, 4$ **and** 7 **is to given in the response sheet. Question No.** $5, 6,$ **and** 8 **is to be given in this booklet itself. All answers to be written in the designated places only. No Additional sheet is allowed.**

3. For each question number to be answered in the response sheet, the last part (i.e. part (u)) is solved to indicate how to answer. Please follow that pattern.

4. All answers to be written using Blue / Black PEN only. **Figures drawn using pencil will not be evaluated.** Page no. 10 may be used for rough work.

5. The compiler which is considered here is the gcc. Answer will be evaluated based on that.

Grade Table (for teacher use only)

Question	Points	Score
1	20	
2	20	
3	20	
4	20	
5	20	
6	20	
7	25	
8	5	
Total:	150	

Student

_____
Signature

Faculty

_____
Signature

1. (20 points) Fill in the blanks with appropriate word/ phrase. A panel of phrases and words with numeric identity is given. You need to write the appropriate number in the response sheet.
(i) bit-wise, (ii) void, (iii) Equality, (iv) Integer, (v) byte addressable, (vi) typedef, (vii) integer type, (viii) data type, (ix) bytes, (x) any order, (xi) Underscore, (xii) String, (xiii) EOF, (xiv) static, (xv) 2, (xvi) 4, (xvii) Macro, (xviii) 3, (xix) no argument, (xx) Float, (xxi) automatic.
   (a) (1 point) #define statement calls for simplest type of $---$ substitution.
   (b) (1 point) The function sprintf places the result in $---$ instead of on the standard output.
   (c) (1 point) The logical operator == checks for $---$ of two values.
   (d) (1 point) The statement "typedef int length" declares length as $---$.
   (e) (1 point) File descriptor is an $---$.
   (f) (1 point) FILE is defined with a $---$ statement.
   (g) (1 point) double xx(void) means that function xx has $---$.
   (h) (1 point) The function getchar() returns $---$ when there is no more input character.
   (i) (1 point) x = f() + g(); In this statement f and g may be evaluated in $---$.
   (j) (1 point) Any pointer can be cast to $---$ without loss of information.
   (k) (1 point) The operator sizeof() return size of a data type in $---$.
   (l) (1 point) To present the use of functions accross different files, $---$ storage class is used.
   (m) (1 point) The expression 11%3 evaluates to $---$.
   (n) (1 point) The contents of two pointers that point to adjacent variables of type float differ by $---$.
   (o) (1 point) ? : is a $---$ operator.
   (p) (1 point) The function that does not return anything has return type $---$.
   (q) (1 point) Every C program automatically opens at least $---$ streams.
   (r) (1 point) The C expression $1 + 2 * 3 - 5$ evaluates to $---$.
   (s) (1 point) An identifier in C must begin with a character or $---$.
   (t) (1 point) Two pointer variables can be compared provided both points to object of same $--$.
   (u) (1 point) fprintf() usually takes $---$ parameters.

2. (20 points) Write the correct option for each of the following question. There is only one correct answer to each question.
   (a) (1 point) Which of the following is not a valid variable name declaration? (a) `int _a3`; (b) `int a_3`; (c) `int 3_a`; (d) `int _3a`;
   (b) (1 point) All keywords in C are in (a) LowerCase letters, (b) UpperCase letters, (c) CamelCase letters, (d) None of the mentioned.
   (c) (1 point) Which of the following is a User-defined data type? (a) typedef int Boolean; (b) typedef enum {Mon, Tue, Wed, Thu, Fri} Workdays; (c) struct {char name[10], int age}; (d) all of the mentioned.
   (d) (1 point) The precedence of arithmetic operators is (from highest to lowest) (a) $\%, *, /, +, -$, (b) $\%, +, /, *, -$, (c) $+, -, \%, *, /$, (d) $\%, +, -, *, /$.
   (e) (1 point) Which of following is true? (a) For loop body always runs once. (b) While loop body always runs once. (c) Do-While loop always runs once. (d) All of the above.

(f) (1 point) Which of the following data type will throw an error on modulus operation? (a) int, (b) float, (c) char, (d) short.

(g) (1 point) Which of the following is equivalent to matrix[i][j]? (a) *((*(matrix)) + (i * COLS + j)); (b) *(*(matrix + i) + j); (c) *(matrix[i] + j); (d) All of the above.

(h) (1 point) If ASCII value of 'x' is 120, then what is the value of the H, if $H = (`x'-`w')/3$; (a)0, (b) 1, (c) 2, (d) 3.

(i) (1 point) Which of the following is not a correct variable type? (a) char (b) real (c) int (d) double;

(j) (1 point) The function that can be used to take the file pointer to the beginning of the file? (a) fopen(), (b) rewind(), (c) fseek(), (d) All the above.

(k) (1 point) About strcat() which option is not true? (a) Appends a copy of the source string to the destination string, (b) Overwrite the null character of the destination string, (c) Destination and source shall not overlap, (d) Not all the above are true.

(l) (1 point) The command to check whether a swap file is created for a file or not is? (a) ls, (b) ls -a, (c) ls -l, (d) ls -t.

(m) (1 point) The header file to include for using malloc() function is? (a) alloc.h, (b) stdio.h, (c) ctype.h, (d) stdlib.h.

(n) (1 point) Which of the following is not true for realloc? (a) Changes the size of the memory block pointed to by ptr, (b) The content of the memory block is always preserved, (c) Behaves similar to malloc if ptr is null, (d) Returns a pointer to its beginning.

(o) (1 point) What is the sequence for preprocessor to look for the file within <> ? (a) The predefined location then the current directory, (b) The current directory then the predefined location, (c)The predefined location only, (d) The current directory location.

(p) (1 point) The "else if" in conditional inclusion is written by ? (a) else if, (b) elseif, (c) elif, (d) elsif.

(q) (1 point) Functions can return enumeration constants in c? (a) True, (b) False, (c) Depends on the compiler, (d) depends on the standard.

(r) (1 point) What is the scope of a function? (a) Whole source file in which it is defined, (b) From the point of declaration to the end of the file in which it is defined, (c) Any source file in a program, (d) From the point of declaration to the end of the file being compiled.

(s) (1 point) Which of following logical operation can be applied to pointers? (Assuming initialization int *a = 2; int *b = 3;) (a) $a|b$, (b) $a\&b$, (c) $a \wedge b$, (d) None of the above.

(t) (1 point) What does this statement printf("%10s", state); means? (a) 10 spaces before the string state is printed, (b) Print empty spaces if the string state is less than 10 characters, (c) Print the last 10 characters of the string, (d) None of the above.

(u) (1 point) Which is the only function all C programs must contain? (a) main(), (b) printf(), (c) scanf(), (d) return().

3. (20 points) There are two sets of phrases/ words given in Table 1. One under the Heading A, and another in B. Each word/ phrase has a number associated with it, eg. A1, B3, etc.

4. (20 points) State whether the following are True/ False.

(a) (1 point) The printf() is a macro that takes variable number of arguments..

(b) (1 point) All macro substitutions in a program happens before the compilation.

(c) (1 point) The function sprintf() is same as fprintf(), but operates on strings, instead of a file.

A	B
1. Modular design, 2. getchar(), 3. int (*p)[10], 4. In fopen() w mode is used, 5. A variable that can hold objects of different type, 6. #include, 7. Static variable, 8. No space is allocated for storage of character during compile time, 9. f(int *a), 10. int *p[10], 11. for(;;), 12. Call by value, 13. calloc(), 14. f(int a), 15. To convert the operand to the specified datatype, 16. rvalue of data item remains unchanged during program execusion, 17. A function that returns object of different type, 18. Actual argument of a function, 19. Address space of data item in memory, 20. Multi level decision. 21. Pointer to a string.	1. Each function should perform one task, 2. Allocate and clear memory, 3. switch statement, 4. must be declared as void, 5. Value parameter in the calling function is unaffected, 6. Function Invocation, 7. Scope till end of file, 8. lvalue, 9. p is a pointer to an array of integers, 10. f(&a[2]), 11. char *p, 12. To create a text file for writting, 13. Returns next characted from stdin, 14. p is pointer to an array of integer pointers, 15. f(a[2]), 16. const type qualifier, 17. Unending loop if no break statement inside the body, 18. Pre-processor command, 19. Must be declared as union, 20. cast operator.

Table 1: Table for Question no 3.

(d) (1 point) Arrays can not be returned by functions, however pointers to arrays can be returned.

(e) (1 point) Members of enumerated datatypes are constants that are written as identifiers, but have signed integer value.

(f) (1 point) A structure has all members of essentially same type.

(g) (1 point) Malloc is a function which is used to allocate memory and returns a pointers to variables.

(h) (1 point) *p++ increments the content of the location pointed by p.

(i) (1 point) Members that compose a union share the same storage area in computer's memory.

(j) (1 point) In a do-while loop an empty while() clause denotes infinite loop, like for(;;).

(k) (1 point) When a function returns a value, the entire function call can appear on the right side of the equality sign.

(l) (1 point) In C programming language strings are represented using an array.

(m) (1 point) We can multiply two characters.

(n) (1 point) A variable can become lvalue and rvalue depending upon the operation.

(o) (1 point) The function fscanf can take variable number of arguments.

(p) (1 point) Constant expressions are evaluated at runtime and not at compile time.

(q) (1 point) Function declaration and function definition are the same thing.

(r) (1 point) Evaluation of the expression (exp1 && exp2) always causes evaluation of exp2.

(s) (1 point) The conversion specifier %*p* is used to print the address.

(t) (1 point) A [I][J] = B + W *[N * (I – Lr) + (J – Lc)], where B = Base address, I = Row subscript of element whose address is to be found, J = Column subscript of element whose address is to be found, W = Storage Size of one element stored in the array (in byte), Lr = Lower limit of row/start row index of matrix, if not given assume 0 (zero), Lc = Lower limit of column/start column index of matrix, if not given assume 0 (zero), M = Number of row of the given matrix, N = Number of column of the given matrix is a column major formulae.

(u) (1 point) Inventor of C language was Dennis Ritchie.

5. (20 points) Write the output of the following code snippets. If there is any error (run time/ compile time), then write what is the error and its reason.

(a) (2 points)
```
#include <stdio.h>
int main(){
  int var = 010;
  printf("%d", var);
}
```

(b) (2 points)
```
#include <stdio.h>
enum birds {SPARROW, PEACOCK, PARROT};
enum animals {TIGER = 8, LION, RABBIT, ZEBRA};
int main(){
  enum birds m = TIGER;
  int k;
  k = m;
  printf("%d\n", k);
}
```

(c) (2 points)
```
#include <stdio.h>
#define MAX 2
enum bird {SPARROW = MAX + 1, PARROT = SPARROW + MAX};
int main(){
  enum bird b = PARROT;
  printf("%d\n", b);
}
```

(d) (2 points)
```
#include <stdio.h>
#include <string.h>
int main(){
  char *str = "x";
  char c = ‘x’;
  char ary[1];
  ary[0] = c;
  printf("%d %d", strlen(str), strlen(ary));
}
```

(e) (2 points)
```
#include <stdio.h>
int main(){
  int x = 1, y = 0, z = 5;
  int a = x && y && z++;
  printf("%d", z);
}
```

(f) (2 points)
```
#include <stdio.h>
```

```
int main(){
  int a = 10, b = 5, c = 5;
  int d;
  d = a == (b + c);
  printf("%d", d);
}
```

(g) (2 points) What is the output of the following C code? Assume that the address of x is 2000 (in decimal) and an integer requires four bytes of memory.

```
int main(){
    unsigned int x[4][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}, {10, 11, 12}};
    printf("%u,%u, %u", x+3, *(x+3),*(x+2)+3);
}
```

(h) (2 points) What will be the value of ?1 and ?2 so that the program below prints an input string in reverse order. Assume that the input string is terminated by a newline character.

```
void reverse(void){
  int c;
  if (?1) reverse() ;
  ?2
}
main(){
  printf ("Enter Text ") ;  printf ("\n") ;
  reverse();  printf ("\n") ;
}
```

(i) (2 points) What is the return value of the function foo when it is called as foo(345, 10)

```
unsigned int foo(unsigned int n, unsigned int r) {
  if (n  > 0) return (n%r +  foo (n/r, r ));
  else return 0;
}
```

(j) (2 points) The value returned by f(1) is

```
int f(int n){
   static int i = 1;
   if (n >= 5)
      return n;
   n = n+i;
   i++;
   return f(n);
}
```

---

6. (20 points) Read the questions carefully and answer to the point.

   (a) (2 points) Can we assign one array to another, give reason(s)?

   (b) (2 points) A character array and a string is not the same thing. Comment?

   (c) (2 points) Writing parameterized Macro is better that writing a function. Comment?

   (d) (2 points) We can successfully compile a .c file without a main(). Comment?

   (e) (2 points) "void value not ignored " a common error. When will you get this error?

   (f) (2 points) User defined data type can be created in C, comment?

   (g) (2 points) Write a recurrence relation for calculating the sum of the digits of a number.

   (h) (2 points) Having a array of pointer to string is a better design, than a $2D$ array of string. Comment.

   (i) (2 points) What is the purpose of the function trial:

```
int Trial (int a, int b, int c) {
```

```
    if ((a > = b) && (c < b)) return b;
    else if (a > = b) return Trial (a,c,b);
    else return Trial (b,a,c);
}
```

(j) (2 points) Write the steps (**no code**) of deleting the k^{th} record from a binary file having n ($n > k$) records.

7. (25 points) Consider the following program. Fill the response sheet accordingly. Consider each box is one cursor location on the screen starting from left to right. Each row signifies a new line. 7(z) is solved for your help. The comment shows the question number.

```
#include<stdio.h>
int main(){
  int n=23; char c='d'; float pi=3.141592654; char *s1 = " a short string";
  printf("%c,\n",c);        /* 7(a) */
  printf("%10c,\n",c);      /* 7(b) */
  printf("%-10c,\n",c);     /* 7(c) */
  printf("%d,\n",n);        /* 7(d) */
  printf("%10d,\n",n);      /* 7(e) */
  printf("%-10d,\n",n);     /* 7(f) */
  printf("%010d,\n",n);     /* 7(g) */
  printf("%4.2d, \n",n);    /* 7(h) */
  printf("%o,\n",n);        /* 7(i) */
  printf("%8o,\n",n);       /* 7(j) */
  printf("%08o,\n",n);      /* 7(k) */
  printf("%-8x,\n",n);      /* 7(l) */
  printf("%f,\n",pi);       /* 7(m) */
  printf("%4.2f,\n",pi);    /* 7(n) */
  printf("%10.4f,\n",pi);   /* 7(o) */
  printf("%16.1f,\n",pi);   /* 7(p) */
  printf("%-16.1f,\n",pi);  /* 7(q) */
  printf("%e,\n",pi);       /* 7(r) */
  printf("%16.1e,\n",s1);   /* 7(s) */
  printf("%s,\n",s1);       /* 7(t) */
```

```
    printf("%4s,\n",s1);       /* 7(u) */
    printf("%-20s,\n",s1);     /* 7(v) */
    printf("%25s,\n",s1);      /* 7(w) */
    printf("%-20.8s,\n",s1);   /* 7(x) */
    printf("%20.8s,\n",s1);    /* 7(y) */
    printf("%06d,\n",n);       /* 7(z) */
}
```

8. (5 points) Draw a flowchart to find the Fibonacci series till $term \leq 1000$.

Rough Work