Eastern Mediterranean University
Department of Computer Engineering

CMPE 112 Final Exam
Spring Semester 2016-2017
June 08, 2017

Name, Surname :…SOLUTION KEY……
Student No :……
Group No :……

Instructors: *Prof. Dr. M. Güler (Gr. 01), Assoc. Prof. Dr. Ekrem Varoğlu (Gr. 02)*

Duration: 100 minutes

Instructions:

- There are **6** questions in this **9** page sheet. Please check !!!!!!
- Calculators are not allowed.
- GSM phones should be turned off.
- A table of operators for precedence and associativity is attached.
- Passing any material including rubbers, pencils etc. to anybody else is strictly prohibited in the exam.
- If an answer box is given in a question, you must give your answer (and nothing else) in the box !!!

Question	Grade
Q1(16 pnts.)	
Q2(16 pnts.)	
Q3(16 pnts.)	
Q4(16 pnts.)	
Q5(16 pnts.)	
Q6(24 pnts.)	
TOTAL: (out of 104)	

PRECEDENCE AND ASSOCIATIVITY

OPERATORS	ASSOCIATIVITY
( ) [ ] -> .	Left to right
! ++ -- + - * & (type) sizeof	Right to left (Unary)
* / %	Left to right
+ -	Left to right
< <= > >=	Left to right
== !=	Left to right
&&	Left to right
ll	Left to right
?:	Right to left
= += -= *= /= %=	Right to left
,	Left to right

Some String Functions:
strlen(s1) Returns length of the string s1
strcat(s1 , s2) Concatenates a copy of string s2 onto the string s1
strncat(s1, s2, n) Concatenates a copy of up to n characters from string s2 onto the string in s1
strcpy(s1, s2) Copies the string s2 to s1
strncpy(s1, s2, n) Copies a string up to **n** characters from s2 to s1
strcmp(s1,s2) Compares the string s2 with the string s1. Returns an integer less than, equal to, or greater than depending on the result of the comparison
strncmp(s1, s2, n) Compares at most the first n characters of the string s1 to s2, and returns an integer less than, equal to , or greater than depending on the result of the comparison
strchr(s1 , c) Locates the first occurrence of c(a char) in the string s1, and returns a pointer to the located character. If the search is not successful than null is returned
strrchr(s1, c) Locates the last occurrence of c(a char) in the string s1, and returns a pointer to the located character. If the search is not successful than null is returned

Q1(16 pnts.):

Fill in the blanks in the program below:

The program reads the names and grades of 30 students from a file called "input.txt". It then decides if a student has passed or failed from the course based on his/her grade. A student passes if the grade is 50 or above. It then writes the names and the result (pass or fail) to another file named "output.txt".

input.txt

```
Ali 40
John 65
Sue 90
Joe 30
... ...
... ...
```

output.txt

```
Ali fail
John pass
Sue pass
Joe fail
... ...
... ...
```

```
#include <stdio.h>
int main (void)
{
 char name[30];
 int grade; i;
 FILE *fp1, *fp2;

 fp1=__fopen("input.txt","r")_________ __________ ; /*open file to read*/

 fp2=___ fopen("ouput.txt","_____________________; /*open file to write*/

 for (__i=0;i<30;i++_____________)

 {

  fscanf(_fp1, "%s %d", &name[i], &grade_____________); /*read from file*/

  if(_grade>=50_____________)

    fprintf(_fp2, "%s pass",_name[i]_____ _________); /*write to file*/

  else

    fprintf(__fp2, "%s fail",_name[i]____  _________); /*write to file*/

 }

 ___fclose(fp1)_____________________;  /* close file */

 ___ fclose(fp2)_____________________;  /* close file */

}
```

Q2(16 pnts.):
Find the output of the following code segments:

a)

```
int k;
char s1[20]= "Engineering", s2[20],s3[20];
strcpy(s3,s1);
printf("%s\n",s3);
strncpy(s1, "abcdefgh",3);
printf("%s\n",s1);
strncpy(s2,&s3[2],3);
strncat(s3, "world",2);
k=strlen(s3);
printf("%s %d\n",s3,k);
printf("%s\n",s2);
```

```
Engineering
abcineering
Engineeringwo13
gin
```

b)

```
char s1[30]="abcde";
char s2[30]="xyz", s3[30];
strcat(s1,s2);
printf("%s\n",s1);
strcpy(s3, strcat(s1,s2));
printf("%s\n",s3);
strncpy(s3, "AAAAAA",6);
printf("%s\n",s3);
strncat(s3,s1,4);
printf("%s\n",s3);
```

```
abcdexyx
abcdexyzxyz
AAAAAAyzxyz
AAAAAAyzxyzabcd
```

Q3(16 pnts.):

Fill in the blanks in the program below:

The program reads 3 exam grades for each of 30 students and finds the average of each exam separately and prints it.

```
#include <stdio.h>
int main (void)
{
   int grades [30][3];
  int i,j,sum;
  float average;
  /*read the grades */
 for ( i=0;i<30; i++___)
 {
   printf ("Enter grades for student %d", i+1)
     for (j=0; j<3; j++__)
        _scanf("%d", &grade[i][j])___;
   }
  /*Find the average of each exam*/
 for (j=0; j<3; j++____)
 {
   __sum=0___;
    for (i=0; i=0;i<30; i++____)
         sum=_sum+ grade[i][j]___;
     average=__sum/30.0___;
    printf ("Average of exam %d is %f", j+1, average);
 }
return 0;
}
```

Q4(16 pnts.):

Give the output of the following program.

```
void fun(char *);
main()
{
 int *pv, v[] = {1,4,7,10,13,16,19};
 char *pc, c[] = "FEDCBA";

 pv = v+4;
 printf("%d  ", (*pv)++);
 printf("%d\n", ++*pv);

 pv = v;
 printf("%d  ", *++pv);
 printf("%d\n", pv[1]);

 fun(c+2);
 printf("%s\n", c);
}

void fun(char *pc)
{
 while(*pc != '\0')
 {
   *pc += 2;
    pc++; }
}
```

```
13   15
4    7
FEFEDC
```

Q5(16 pnts.):

Give the output of the following program.

```
#include <stdio.h>

main()
{
  int x = 0, z = -7;
  do
  {
    z++;
    if(x)
    {
      printf("How \n");
      if(z+3 == x) continue;
       printf("Are you \n");
    }
    x -= 2;
    printf("Today ? \n");
  }while(z < -4);
}
```

```
Today ?
How
How
Are you
Today ?
```

Q6(24 pnts.):

Give the outputs of the following programs.

a)

```
int suba(int *xp, int av)
{
    if (*xp < av)    *xp += av;
                else   *xp  -= av;
    av += 3;
    printf("Result= %d  %d\n", *xp , av);
    return av;
}

main()
{
   int x , y;
   x=2, y=7;
   x += suba(&x , y);
   printf("X= %d  Y= %d",x ,y);
}
```

Result= 9 10
X= 19 Y= 7

//continues on the next page !!

b)

```
 main()
{
  int *fun(int *, int);
  int v[]={1,3,5,7,9}, *q;

  q = fun(v,2);
  printf("%d\n",q[1]);
}

int *fun(int *p, int a)
{
  return p+a;
}
```

7

c)

```
 #include <stdio.h>

  main()
  {
    char *p = "ABCDEF";
     printf("%c\n",  *(p + *p - 'C' + 5));
     printf("%s\n",  p + 3);
  }
```

D
DEF