CMPE/CMSE/AING 112

Programming Fundamentals

FINAL Examination (June 15th, 2024)

Duration - 110 minutes | [12:30 – 14:20]

Total Number of Questions: 35

Name, Surname _____

STUDENT ID _____

GROUP NUMBER _____

Department: (Circle your department appropriately)

CMPE / CMSE / AING

Gr.2 Prof. Dr. Marifi Güler

Gr.3 Prof. Dr. Ekrem Varoğlu

Gr.5 Prof. Dr. Doğu Arifler

Gr. 6 Prof. Dr. Önsen Toygar

Instructions:

- The exam is composed of **35** questions on **21** pages.
- Calculators are not allowed.
- Smart watches are not allowed.
- A table of operators for precedence and associativity, and a list of some popular C string functions is attached.
- Passing any material including rubbers, pencils etc. to anybody else is strictly prohibited during the exam.
- Wearing hats/baseball caps etc. is not allowed.
- Mark your answers on the optic form provided.
- ONLY THE OPTIC FORMS WILL BE GRADED.
- Return BOTH your question booklet and the optic form at the end of the examination.

PRECEDENCE AND ASSOCIATIVITY

OPERATORS	ASSOCIATIVITY
( ) [ ] -> .	**Left to right**
! ++ -- + - * & (type)	**Right to left (Unary)**
*** / %**	**Left to right**
+ -	**Left to right**
< <= > >=	**Left to right**
== !=	**Left to right**
&&	**Left to right**
||	**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 strings s1 and s2, returning a value less than, equal to, or greater than 0, depending on whether s1 is less than, equal to, or greater than s2.

1. Given the following declarations/initializations at the beginning of a C function:

   char str1[ ] = "Hello", str2[20] = "Hi";

   Which of the following statements within the same scope **will not** give an error?

   (a) str1 = "Bye";
   (b) str1 = str2;
   (c) str1++;
   (d) *str1 = 'Z';

2. Given the following declarations/initializations at the beginning of a C function:

   char *p = "Hello", *s = "Hi";

   Which of the following statements within the same scope will give an error?

   (a) *p = 'M';
   (b) p = "Bye";
   (c) p = s;
   (d) p++;

3. What is the output of the following C program fragment?

```
char s[ ] = "CMPE" ;
int i = 0 ;
while ( s[ i ] != 0 ) {
printf ( "%c", *( s + i++ ) + 1 ) ;
}
```

   (a) CMPE
   (b) DNQF
   (c) EPMC
   (d) EORG

4. What is the output of the following C program fragment?

```
int x = 10 ;

void f() {
   printf ("%d ", x ) ;
}

int main( ) {
   int x = 20 ;
   {
      int x = 30 ;
      printf ("%d ", x ) ;
   }
   printf ("%d ", x ) ;
   f();
   return 0 ;
}
```

(a) 10 20 30
(b) 30 20 10
(c) 30 10 20
(d) 20 30 10

5. What is the output of the following C program fragment?

```
void f( ) {
   int i = 0 ;
   static int j = 0 ;
   i++ ; j++ ;
   printf ("%d %d ", i, j) ;
}

int main( ) {
   f( );
   f( );
   f( );
   return 0 ;
}
```

(a) 1 1 1 2 1 3
(b) 1 1 1 1 1 1
(c) 1 2 1 2 1 2
(d) 1 2 2 3 3 4

6. What is the output of the following C program fragment?

```
void f(int *p, int x) {
   *p += 5;
   x += 5;
}

int main() {
   int a = 4;
   int b[]={10, 20};
   f(&a, b[1]);
   printf("a = %d b[1] = %d\n", a, b[1]);
   return 0;
}
```

(a) a = 9 b[1] = 25
(b) a = 5 b[1] = 20
(c) a = 9 b[1] = 10
(d) a = 9 b[1] = 20

7. What is the output of the following C program fragment?

```
   int k, *p, v[] = {5, 6, 7, 8};

   p = v + 2;
   p[-1] = 10;
   *(v+3) = *p + 5;
   for (k = 0; k < 4; k++) printf("%d  ", v[k]);
```

(a) 5 11 10 8
(b) 7 6 11 8
(c) 5 10 7 12
(d) Program gives an error

8. What is the output of the following C program fragment?

```
void pb(int n) {
   putchar('A' + n / 2);
   putchar('A' + n % 2);

}

int main () {
   pb(9);
    return 0;
}
```

(a) EB
(b) FB
(c) 41
(d) 0401

9. What is the output of the following C program fragment?

```
char s1[50], s2[50], s3[50];
strcpy(s1, "computer ");
strcpy(s2, "engineering ");
strcpy(s3, "love ");
if (strcmp(s2, s1) < 0)
   strcat(s3, strcat(s1, s2));
else
   strcat(s3, strcat(s2, s1));
     printf("%s", s3);
```

(a) love computer engineering
(b) engineering computer love
(c) love engineering computer
(d) computer engineering love

10. Consider the following C program and find the output:

```
#include<stdio.h>
int main(void)
{    double a[ ] = {3.9, 2.4, 7.7, 1.1};
     double *p;
     p = a;
     p++;
     printf("%5.2f  ", *p);
     printf("%5.2f  ", p[1]);
     printf("%5.2f  ", a[0]);
     return (0);
 }
```

a) 3.90 2.40 7.70
b) 2.40 7.70 3.90
c) 2.40 2.40 3.90
d) 3.90 7.70 3.90

11. What will be printed by the following C code fragment?

```
char *p="FINAL";
printf("%c\n", *(p+'G'-*p+2));
```

a) F b) I c) N d) A

12. Trace the following program and find what will be stored in string d4 at the end of the following C code fragment:

```
char d1[20] = "ARTIFICIAL", d2[20]="COMPUTER";
char d3[20] = "SOFTWARE", d4[20];
strcpy(d2, d1);
strncpy(d4, "ENGINEERING", 8);
strncpy(d1, &d3[4], 3);
strncat(d4, d1, 3);
```

a) WARIFICIAL b) ARTIFICIAL c) ENGINEERWAR d) ENGINEERINGWAR

13. What will be the values of array elements after calling “Magic” function?

```
#include<stdio.h>
void Magic(int *p)
{ ++p;
  *p = *p+1;
  p++;
  *p=*p-1;
}
int main(void)
{    int a[5] = {8, 12, 6};
     Magic(a);
     return (0);
 }
```

a) 8 13 5 0 0
b) 8 12 5 0 0
c) 9 11 6 0 0
d) 9 12 6 0 0

14. What will be the output of the following C code fragment?

```
#include<stdio.h>
int main(void)
{    int m, n,  *p, a[5] = {10, 12, 14, 16, 18};
     p = a;
     n =  ++*p +2;
    m = *p++ + 3;
    printf(“%d  %d\n”, n, m);
    return (0);
 }
```

a) 13 13
b) 13 14
c) 14 14
d) 14 15

15. Given the following initializations, which of the following expressions will evaluate to zero?

    int numbers[] = {1, 0, 3, 0, 5};
    int *q = numbers;

    a) *(q+2)
    b) --*(q+1)
    c) q[2]-2
    d) q-numbers

16. What should be written into the empty space in the program in order to print the second row of the array three times as the output of the program?

    Output:
    4 5 6
    4 5 6
    4 5 6

```
#include <stdio.h>
int main()
{ int i, j;
 int  t[3][3] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
for (i = 0; i < 3; i++)
    { for (j = 0; j < 3; j++)
          printf("%d  ", ……..);
       printf(“\n”);}
 return 0;
 }
```

a) t[i][j] b) t[1][j] c) t[i][1] d) t[1][1]

17. What will be the output of the following C code fragment?

```
int i, sum=0;
for (i=1; i<=3; i++)
    { static int st=0;
      sum = sum + st + i;
      st++;
    }
 printf("%d\n", sum);
```

a) 3 b) 6 c) 7 d) 9

18. Which one of the following functions is used to read one character in C language?

a) strlen()

b) strcmp()

c) getchar()

d) gets()

19. Consider the function fun:

```
float fun(float *p, int n)
{
   int k;
  float a=0.0;
  for(k=0;k<n;k++)
   if(p[k] > 0.0)  a+= p[k];
   else a - = p[k];
 return a;
}
```

What does the function do when it is called as follows?

```
        float arr[100];
    …
printf("%.2f\n", fun(arr+1, 100));
```

a) It prints the sum of absolute values of the elements of **arr[100]**

b) It prints the sum of the positive elements of **arr[100]**

c) It prints the sum of absolute values of the elements of **arr[100]**, except the first element

d) It prints the sum of the positive elements of **arr[100]**, except the first element

20. Consider the function fun:

```
void fun(int *pt, int n)
{
 int *q;
 n--;
 for(_____ MISSING CODE_____)     printf("%d   ",  *q);
 printf("\n");   }
```

However, part of the code is missing (indicated by _____). The function is supposed to print the values of an array in reverse order (without changing the contents of the array).

For example, when it is called as

```
 int array[5] = {1, 2, 3, 4, 5};

 fun(array,sizeof(array)/sizeof(int));
```

it will give the output **5 4 3 2 1**

What can the missing part be?

a) **q=pt+n; q >= pt; q**--

b) ***q=*pt+n; *q >= *pt; *q**--

c) **q=pt+n; q >= pt; *q**—

d) ***q=*pt+n; *q >= *pt; (*q)**—

21. Consider the code

```
char s[100], news[100]={'\0'};
int i=0, k=0;
gets(s);
if(strlen(s) > ____Missing_1___  )
{
 printf("Too long string\n");
 return;
}
for(; s[i] != '\0'; i++) {
  if(s[i] == ' ') ____Missing_2___ ;
  news[ ____Missing_3___] = s[i];}
printf("%s\n", news);
```

However, part of the code is missing (indicated by _____). The code is supposed to copy the contents of the array **s** to **news** without the blanks (e.g. **" He is fine"** is copied as **"Heisfine"**).

What can the missing parts be?

a) Missing_1: **100** Missing_2: **continue** Missing_3: ++**k**

b) Missing_1: **99** Missing_2: **break** Missing_3: **k++**

c) Missing_1: **100** Missing_2: **break** Missing_3: ++**k**

d) Missing_1: **99** Missing_2: **continue** Missing_3: **k++**

22. Consider the code

```
int k;
char c[] = "ABCDEFG";
char *cp;
for(cp = c+2; cp >= c;) printf("%c", ____Missing_1___ );
printf("\n");
for(cp = c+3; cp <= c+6;) printf("%c", ____Missing_2___ );
```

However, part of the code is missing (indicated by _____). The code is supposed to give the output

```
CBA
EFG
```

What can the missing parts be?

a) Missing_1: ***--cp** Missing_2: ***++cp**

b) Missing_1: ***cp--** Missing_2: ***++cp**

c) Missing_1: (***cp)--** Missing_2: (***cp)++**

d) Missing_1: (***cp)--** Missing_2: ++(***cp)**

23. Consider the declarations:

float x[4][7];

float y[28];

char s[] = “ibm”;

char q[] = {‘i’, ‘b’, ‘m’};

int v[5] = {1,2};

Which of the following is NOT correct?

a) Both **x** and **y** have **28** elements and occupy the same amount of memory

b) **v** has **5** elements

c) **s** has **3** elements

d) **q** has **3** elements

24. Suppose we want to declare an array of characters to hold a C string with exactly 9 letters. Which declaration is best?

a) char s[8];

b) char s[9];

c) char s[10];

d) char s[11];

25. Given the code

```
char *ptr;
char myString[]="abcdefg123ABCDEFG";
ptr = myString + sizeof(myString);
ptr -= 3;
```

The pointer ptr points to string:

a) **"ABCDEFG"**

b) **"123"**

c) **"FG"**

d) **"d"**

26. What is the output of the following program code?

```
void fun(int*, int);
main()
{
  int x[2]={0,1};
  fun(&x[0], x[1]);
  printf("%d  %d\n", x[0], x[1]);
}
void fun(int *a, int b)
{
  *a += ++b;
  b  += ++*a;
}
```

a) 3 1 **b) 4 1** **c) 3 2** **d) 3 3**

27. Consider the following declaration:

char s1[] = "AB", s2[] = "CDF", s3[10]= "Z";

The correct way to have the string **"ZABCDF"** in s3 is

a) strcpy(s3,s1); strcat(s3,s2);

b) strcpy(s3+1,s1); strcat(s3,s2);

c) strcpy(s3,s1); strcat(s3+1,s2);

d) strcpy(s3+1,s1); strcat(s3+1,s2);

28. What is the sizeof(char) in a 32-bit C compiler?

a) 1 bit
b) 2 bits
c) 1 Byte
d) 2 Bytes

29. What is the output of this C code?

```
#include <stdio.h>
  main()
  {
     int i = 0, j = 0;
     if (i > 0)
        if (j > 0)
           printf("1");
     else
        printf("2");
  }
```

a) 1
b) 2
c) No output will be printed
d) Run time error

30. How many times is the test value checked in the following C program?

```
#include <stdio.h>
int main()
{
   int test = 0;
   while (test < 100)
      test++;
   printf("enjoy\n");
}
```

a) 99
b) 100
c) 101
d) cannot be determined

31. What will be the data type of the result of the following operation?

```
(float)a * (int)b / (long)c * (double)d
```

a) int
b) long
c) float
d) double

32. What will be the output of the following program?

```
#include <stdio.h>
 void main()
{
  f();
void f()
{
        printf("I love C");
 }
}
```

a) I love C
b) Compile time error
c) The program will not print anything
d) Depends on the input

33. What will be the output of the following C code?

```
 #include <stdio.h>
 void main()
{
 char *smile = "hello";
char *new = "cjn";
char *plan = smile + new;
 printf("%c  %c", *plan, smile[1]);
}
```

a) h e
b) Compile time error
c) c o
d) h n

34. What will be the output of the following C code?

```
#include <stdio.h>
   void f(int *ptr)
   {
      int i = 0;
      for(i = 0;i < 4; i++)
      printf("%d", ptr[i]);
   }
int main()
{
  int arr[5] = {1, 2, 3};
      f(&arr[1]);
   return 0;
}
```

a) 0000
b) 2300
c) Compile time error
d) 1230

35. Given the following 2 programs:

Program1:	Program2:

#include <stdio.h>
int x = 5;
void main()
{
int x = 3;
printf("%d", x);
{
x = 4;
}
printf("%d", x);
}
	#include <stdio.h>
	int x = 5;
	void main()
	{
	int x = 3;
	printf("%d", x);
	{
	int x = 4;
	}
	printf("%d", x);
	}

What will be the output of the two programs respectively?

a) Program1: 3 4 Program2: 3 3

b) Program1: 3 4 Program2: 3 4

c) Program1: 3 5 Program2: 3 5

d) Program1: 3 5 Program2: 4 5