# National University of Computer & Emerging Sciences, Karachi

**Fall-2020 CS / SE / AI / DF Department**

**Midterm 2**

**23rd November 2020, 12:00 noon – 01:00 pm**

| | |
|---|---|
| **Course Code: CS118** | **Course Name:Programming Fundamentals** |
| **Instructor Name: M. Shahzad/ Basit Ali / S. Zain / Atiya / Musawar / Nida** | |
| **Student Roll No:** | **Section No:** |

## Instructions:

- Return the question paper and make sure to keep it inside your answer sheet.
- Read each question before answering it. There are **4 questions and 2 pages**.
- In case of any ambiguity, you may make assumption. But your assumption should not contradict any statement in the question paper.
- You are **not allowed to write** anything on the question paper (except your ID and group).
- **Do not write any header files in writing your solution.**

**Time**: 60 minutes. **Max Marks**: 40 Points

**Q1.** Observe and try to understand the following programs. Write the output of the program after correction (if needed): **[10 mins, 10 points]**

**(i).**

```
void message( )
{
printf ( "\nPraise worthy and C
worthy are synonyms" ) ;
}
void main()
{
message(message ()) ;
}
```

**(ii).**

```
main( )
{
int three[3][ ] = {
2, 4, 3,
6, 8, 2,
2, 3 ,1
} ;
   printf ( "\n%d", three[1][1]
   ) ;
}
```

**(iii).**

```
  main( )
     {
       int i = 1, j = 1;
       for ( ; ; )
           {
           if ( i > 5 )
           break;
           else
           j += i;
           printf ( "\n%d", j );
           i += j;
                 }
     }
```

**(iv).**

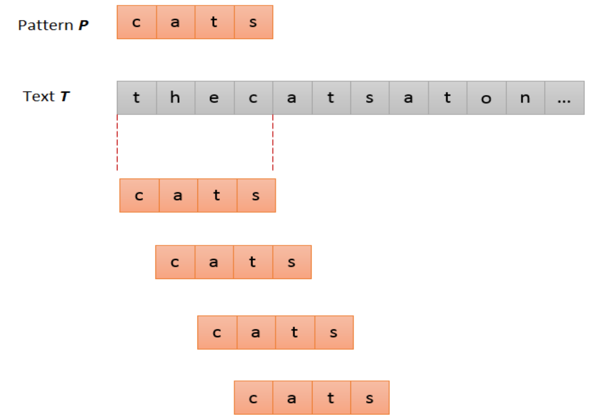
```
main( )
    {
    int i = 4, j = 2 ;
    junk ( &i, j ) ;
    printf ( "\n%d %d", i, j ) ;
    }
    junk ( int *i, int j )
    {
    *i = *i * *i ;
    j = j * j ;
    }
```

**(v).**

```
main( )
{
char s[ ] = "Get organised! learn C!!" ;
printf ( "\n%s", &s[2] ) ;
printf ( "\n%s", s ) ;
printf ( "\n%s", &s ) ;
printf ( "\n%c", s[2] ) ;
}
```

**Q2.** Write a C program that implements the given algorithm. Your program read a large text ***T*** of length ***N*** and pattern ***P*** of length ***M*** such that ***M < N***. Compare pattern ***P*** to each of substring of text ***T*** of length ***M*** and display the message "Pattern Found" if pattern ***P*** exist in the text ***T,*** else display message "Not Found". **[15 mins, 10 points]**

**Example Input/Output:** **Algorithm**



```
do
  if (text letter == pattern letter)
    compare next letter of pattern to next
    letter of text
  else
    move pattern down text by one letter
while (entire pattern found or end of text)
```

**Q3.** Consider a 2-dimensional 3 x 3 array. Write a complete program to achieve the following: **[20 mins, 1+3+3+3 points]**

I. Populate this array with values taken from user
II. Find sum of each row in the array & also find the maximum sum among them
III. Display maximum sum & the particular row containing that sum
IV. Replace all values in every row with the maximum sum except for the row containing that sum

**Q4.** Design a recursive function with the prototype "***int rec(int arr[], int size)***" in C, that traverse complete integer array recursively and return total number of negative values. **[10 mins, 10 points]**

*********** Best of Luck***********