1. Write a recursive function \( \text{FACT}(n) \) to calculate the factorial of \( n \) where \( n \) will be provided from the keyboard. Clearly show in border the recursive statements and the termination conditions.  

2. What are the disadvantages of Recursion?  

3. Write a program that declares a 2D matrix MAT\([N][N]\) where \( N \) is 6 and using loops, fills the matrix with the following values.

\[
\begin{array}{cccccc}
1 & 2 & 3 & 4 & 5 & 6 \\
7 & 7 & 7 & 7 & 7 & 7 \\
8 & 9 & 10 & 11 & 12 & 13 \\
14 & 14 & 14 & 14 & 14 & 14 \\
15 & 16 & 17 & 18 & 19 & 20 \\
21 & 21 & 21 & 21 & 21 & 21 \\
\end{array}
\]

4. Find the exact output from the following code segment. The current memory status for the program is shown on the right side of the program.

```c
int p, s, *ptr;
int a[3] = {2, 5, 7};
ptr = a;
printf("%d\n", *(ptr+1) );
*ptr = *ptr + 1;
printf("%d\n", *ptr);
ptr = s;
printf("%d\n", *ptr);
ptr --;
printf("%d\n", *(ptr));
printf("%s", &s);
```

<table>
<thead>
<tr>
<th>Address</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFA0</td>
<td>p</td>
</tr>
<tr>
<td>FFA1</td>
<td>12</td>
</tr>
<tr>
<td>FFA2</td>
<td>s</td>
</tr>
<tr>
<td>FFA3</td>
<td>14</td>
</tr>
<tr>
<td>FFA4</td>
<td>a</td>
</tr>
<tr>
<td>FFA5</td>
<td>2</td>
</tr>
<tr>
<td>FFA6</td>
<td>5</td>
</tr>
<tr>
<td>FFA7</td>
<td></td>
</tr>
<tr>
<td>FFA8</td>
<td>7</td>
</tr>
<tr>
<td>FFA9</td>
<td></td>
</tr>
<tr>
<td>FFAB</td>
<td></td>
</tr>
<tr>
<td>FFAC</td>
<td>ptr</td>
</tr>
</tbody>
</table>
1. Write a recursive function \texttt{POWER(x, n)} to calculate \(x^n\) where \(x\) and \(n\) will be provided from the keyboard. Clearly show in border the recursive statements and the termination conditions. 3

2. What are the advantages of Recursion? 1

3. Write a program that declares a 2D matrix \texttt{MAT[N][N]} where \(N\) is 5 and using loops, fills the matrix with the following values.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>15</td>
<td>20</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>16</td>
<td>22</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>17</td>
<td>24</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>18</td>
<td>26</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>14</td>
<td>19</td>
<td>28</td>
<td>33</td>
<td></td>
</tr>
</tbody>
</table>

4. Find the exact output from the following code segment. The current memory status for the program is shown on the right side of the program. 3

```c
int b=10, c=15, *ptr;
int a[2]= {1,5};
ptr = &b;
printf("%d\n", *ptr);
ptr++;  
printf("%d\n", *ptr);
ptr = a;
printf("%d\n", *(ptr+1));
*ptr = *(ptr) - 1;
printf("%d\n", *(ptr+1));
printf("%x", &b);
```

```
FF10 FF11 FF12 FF13 FF14 FF15 FF16 FF17 FF18 FF19 FF1A FF1B FF1C FF1D
  10 a b  c
  FF10 FF11 FF12 FF13 FF14 FF15 FF16 FF17 FF18 FF19 FF1A FF1B FF1C FF1D
  FF10 FF11 FF12 FF13 FF14 FF15 FF16 FF17 FF18 FF19 FF1A FF1B FF1C FF1D
  FF10 FF11 FF12 FF13 FF14 FF15 FF16 FF17 FF18 FF19 FF1A FF1B FF1C FF1D
  FF10 FF11 FF12 FF13 FF14 FF15 FF16 FF17 FF18 FF19 FF1A FF1B FF1C FF1D
  FF10 FF11 FF12 FF13 FF14 FF15 FF16 FF17 FF18 FF19 FF1A FF1B FF1C FF1D
  FF10 FF11 FF12 FF13 FF14 FF15 FF16 FF17 FF18 FF19 FF1A FF1B FF1C FF1D
  FF10 FF11 FF12 FF13 FF14 FF15 FF16 FF17 FF18 FF19 FF1A FF1B FF1C FF1D
  FF10 FF11 FF12 FF13 FF14 FF15 FF16 FF17 FF18 FF19 FF1A FF1B FF1C FF1D
  FF10 FF11 FF12 FF13 FF14 FF15 FF16 FF17 FF18 FF19 FF1A FF1B FF1C FF1D
  FF10 FF11 FF12 FF13 FF14 FF15 FF16 FF17 FF18 FF19 FF1A FF1B FF1C FF1D
  FF10 FF11 FF12 FF13 FF14 FF15 FF16 FF17 FF18 FF19 FF1A FF1B FF1C FF1D
  FF10 FF11 FF12 FF13 FF14 FF15 FF16 FF17 FF18 FF19 FF1A FF1B FF1C FF1D
```