**Expr 10 a: Best Fit**

**First Fit code:**

#include &lt;stdio.h&gt;

#define MAX 10

int main() {

int blockSize[MAX], fileSize[MAX], allocation[MAX], originalBlockSize[MAX];

int blockCount, fileCount, fragment[MAX];

// Input number of blocks and files

printf(&quot;Enter the no of blocks: &quot;);

scanf(&quot;%d&quot;, &amp;blockCount);

printf(&quot;Enter the no of files: &quot;);

scanf(&quot;%d&quot;, &amp;fileCount);

// Input block sizes

printf(&quot;Enter the size of the blocks:\n&quot;);

for (int i = 0; i &lt; blockCount; i++) {

printf(&quot;Block %d: &quot;, i + 1);

scanf(&quot;%d&quot;, &amp;blockSize[i]);

originalBlockSize[i] = blockSize[i]; // Keep original size for output

}

// Input file sizes

printf(&quot;Enter the size of the files:\n&quot;);

for (int i = 0; i &lt; fileCount; i++) {

printf(&quot;File %d: &quot;, i + 1);

scanf(&quot;%d&quot;, &amp;fileSize[i]);

allocation[i] = -1; // Initially not allocated

fragment[i] = -1; // Initially no fragment

}

// First Fit Allocation

for (int i = 0; i &lt; fileCount; i++) {

for (int j = 0; j &lt; blockCount; j++) {

if (blockSize[j] &gt;= fileSize[i]) {

allocation[i] = j;

fragment[i] = blockSize[j] - fileSize[i];

blockSize[j] -= fileSize[i]; // Update available size

break;

}

}

}

// Final output

printf(&quot;\nFile No\tFile Size\tBlock No\tBlock Size\tFragment\n&quot;);

for (int i = 0; i &lt; fileCount; i++) {

printf(&quot;%d\t%d\t\t&quot;, i + 1, fileSize[i]);

if (allocation[i] != -1) {

int b = allocation[i];

printf(&quot;%d\t\t%d\t\t%d\n&quot;, b + 1, originalBlockSize[b], fragment[i]);

} else {

printf(&quot;Not Allocated\t-\t\t-\n&quot;);

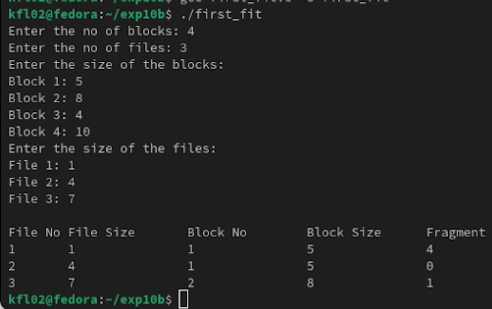
}

}

return 0;

}

**Output:**



**Result:**

Thus the First fit Code is implemented in fedora using the c language