

```

def naive_string_matching(text, pattern):
    """
    Finds all starting indices in the text where the pattern matches exactly.

    Args:
    text (str): The input text string of length n.
    pattern (str): The pattern string of length m.

    Returns:
    list: A list of starting indices where the pattern matches the text.
    """
    n = len(text)
    m = len(pattern)
    result = []

    # Edge case: If the pattern is empty, return an empty list
    if m == 0:
        return result

    # Iterate through the text to find matches
    for i in range(n - m + 1): # Loop until the last possible starting index
        # Check if the substring matches the pattern
        if text[i:i + m] == pattern:
            result.append(i) # Add the starting index to the result list

    return result

# Example usage
if __name__ == "__main__":
    # Input text and pattern

```

```
text = "THIS IS A TEST TEXT"
```

```
pattern = "TEST"
```

```
# Find all starting indices of the pattern in the text
```

```
matches = naive_string_matching(text, pattern)
```

```
# Output the result
```

```
if matches:
```

```
    print(f"Pattern found at indices: {matches}")
```

```
else:
```

```
    print("Pattern not found.")
```