flowchart

```
start
```

```
def arrange_scores()
```

```
open("testfile.txt", "r+") as Score_Manager
```

```
Variables:
lines  , lined  , header
```

```
lines = Score_Manger.readlines()
lined =lines[3:]
header = lines[:3]
```

```
lined.sort(key=lambda line: int(line.split('|')[1].strip()), reverse=True)
```

```
open("testfile.txt", "w") as Score_Manager
```

```
Score_Manager.writelines(header)
```

```
Score_Manager.writelines(lined)
```

```
def score(new_name, new_score)
```

```
open("testfile.txt", "a+") as Score_Manager
```

if new_score < 0 — **true** → Display "Score starts from 0" → return (end the excution of the function)

```
open("testfile.txt", "a+") as Score_Manager
```

```
Score_Manager.seek(0)
```

```
Variables:
contents
```

```
contents = Score_Manager.read()
```

If not contents → **true** →
```
Score_Manager.write("High
scores manager\n".center(75))
```

```
Score_Manager.write("\n player" + "|".center(24)
+ "Top scores\n".center(10))
```

false

```
Variables:
lines
```

```
Lines = [line.strip() for line in contents.split('\n')]
```

for line in lines

Check all lines

```
Variables:
Parts , name, , new_name_stripped
```

```
parts = line.split('|')
```

```
name = parts[0].strip()
```

new_name_stripped = new_name.lstrip()

if new_name_stripped == name

true → Display " The name exists"

return (end the excution of the function)

false

Score_Manager.write(f"{new_name:<17} | {new_score:^30}\n")

return (end the excution of the function)

Display "Hello our user to high score manager"

Display "Please choose your preferred option"

open("testfile.txt", "a+") as Score_Manager

Score_Manager.close()

While True

Variables:
option

Display (add scores |1| || search |2| update |3| || exit |4|) : ")

Save input in option

If option
==
"1"

fasle

true

Variables:
name , user_score

Display "Enter your name: "

Save input
in name

"Score with numbers only"

Display "Enter your score: "

except
ValueError

Save input in
user_score

Error input

try if name == ""
or
user_score == ""

true

Display " Empty inputs
are invalid "

false

score(name, int(user_score))

arrange_scores()

elif option == "2"

fasle

true

Display "Searching for scores....."

Variables:
search_name , found

Display "Enter the player's name without unnecessary spaces : "

Save input in search_name

found = False

open("testfile.txt", "r+") as Score_Manager

for line in Score_Manager

Check all lines

Variable:
parts

parts = line.split('|')

```
if search_name
==
parts[0].strip():
```

false

true

Display (line)

found = True

```
if not found
or
search_name == "":
```

"No matching names found"

```
elif option
==
"3"
```

false

true

Display "Updating scores......"

Variables:
update_name
new_score1

Display "Input player name for score update without extra spaces : "

Save input in update_name

Display "Enter the new score: "

Save input in
new_score1

if new_score1 == ""
or
update_name == "":

true → Display "Empty inputs are invalid"

false

elif new_score1.isalpha()
or
not new_score1.isnumeric():

true → Display "Score with positive numbers only"

false

Variables:
scored, updated_lines
score_exists1

scored = []
updated_lines = []
score_exists1 = False

open("testfile.txt", "r") as Score_Manager

Variables:
lines

lines = Score_Manager.readlines()

```
for line in lines
```

Check all lines

```
Variables:
parts
```

```
parts = line.split('|')
```

```
if parts[0].strip()
==
update_name
```

false

true

```
scored.append(line)
```

```
score_exists1 = True
```

```
updated_line = line.replace(f"
{parts[1].strip()} ", f" {new_score1} ")
```

```
updated_lines.append(updated_line)
```

```
"Score successfully updated")
```

```
if not
score_exists1
```

true

```
print("Name not found")
```

false

```
open("testfile.txt", "w") as Score_Manager
```

```
Score_Manager.writelines(scored + updated_lines)
```

```
arrange_scores()
```

elif option
==
"4"

true

Display "Thank you for using the high score manager "

Display "Note, the file saved any changes"

Exit from the program

false

Display "Choose from available options. Try again"