

Write a C-program to simulate the following contiguous mem. alloc. techniques 1) Best fit 2) first fit 3) Worst fit.

### Best fit

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
#define max 25
```

```
void main()
```

```
{ stores frag. values      blocksize      file      no. of blocks .
```

```
int frag[Max], b[max], f[max], i, j, nb, nf, temp,
```

```
lowest = 10000;      block usage      file alloc .
```

```
static int bf[max], ff[max];
```

```
else { i++; }
```

```
printf("Enter the no. of blocks: ");
```

```
scanf("%d", &nb);
```

```
printf("Enter the no. of files: ");
```

```
scanf("%d", &nf);
```

```
printf("Enter the size of blocks: \n");
```

```
for (i=1; i<=nb; i++)
```

```
{}
```

```
printf("BLOCK %d", i);
```

```
scanf("%d", &b[i]);
```

```
}
```

```
printf("Enter size of files: \n");
```

```
for (i=1; i<=nf; i++)
```

```
{}
```

```
printf("FILE %d", i);
```

```
scanf("%d", &f[i]);
```

```
}
```

```
for (j=1; j<=nb; j++)
```

```
{}
```

```
if (bf[j] == 0) → checks if block is not already used.
```

```
{}
```

```
temp = b[j] - f[i];
```

```

if (temp >= 0) non-neg.
if (lowest > temp)
    $
```

```

if (P[i] = j);
lowest = temp;
$
```

```

frag[i] = lowest;
bt[ffff[i]] = 1;
lowest = 1000;
```

```

printf ("\\n file-no \\t file-size \\t block-no \\t block-size
\\t fragment");
for (i = 1; i < n; i++)
    if (bt[i] == 0; i++)
        getch();
```

\$

### Output

Enter the number of process : 2

Enter the number of block : 3

Enter the process sizes : 1 4 3 4 1 1 1 1 1 1

Enter the block size : 5 2 7

Process No.	Process size	Block no
1	1	1
2	4	1 2 3

1

2

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

first fit

```
#include <stdio.h>
#include <conio.h>
#define max 25
void main()
```

{

```
int frag[max], b[max], i, j, n, b, nf, temp;
```

```
static int bf[max], ff[max];
```

```
clrscr();
```

```
printf("Enter no. of blocks\n");
```

```
scanf("%d", &n);
```

```
printf("Enter no. of files\n");
```

```
scanf("%d", &nf);
```

```
printf("\nEnter the size of blocks\n");
```

```
for (i = 1; i <= n; i++)
```

```
printf("Block %d", i);
```

```
scanf("%d", &b[i]);
```

```
}
```

```
printf("Enter the size of files:\n");
```

```
for (i = 1; i < nf; i++)
```

```
{
```

```
printf("file %d", i);
```

```
scanf("%d", &f[i]);
```

```
}
```

```
for (i = 1; i < nf; i++)
```

```
{
```

```
for (j = 1; j <= n; j++)
```

```
{
```

```
if (bf[j] == 1)
```

```
{
```

```
temp = b[j] - f[i];
```

```
if (temp >= 0)
```

```
{
```

```
bf[i] = j;
```

```
break;
```

```
{
```

```
}
```

```
frag[i] = temp;
```

```
bff[ffff[r]] = 1
```

```
g. .... { i. [xam]. [xam] + .xam put + }
```

```
printf("In file-no: %d file-size: %d Block-no: %d  
size: %d fragment");
```

```
for (i=0; i < nf; i++)
```

```
printf("%d.%d%d%d/%d%d", i, tf[i], ff[i], bff[i]);
```

```
frag[i];
```

```
getch();
```

4.

### Output

Enter the num. of blocks: 3

Enter the num. of files: 2

Enter the size of blocks

Block 1: 5

Block 2: 2

Block 3: 7

Enter the size of files

file 1:

file 2: 4

file-no	file-size	Block-no	Block-size
1	1	1	5
2	4	3	7

worst fit

```
#include <stdio.h>
#include <conio.h>
#define max 25
void main()
{
    int frag[max], b[max], f[max], i, j, n, b, nf, temp,
        highest = 0;
    static int bf[max], ff[max];
    clrscr();
    printf("\nEnter the no. of blocks :");
    scanf("%d", &n);
    printf("\nEnter the number of files:");
    scanf("%d", &nf);
    printf("\nEnter size of blocks :\n");
    for (i = 1; i <= n; i++)
    {
        printf("Block %d ", i);
        scanf("%d", &b[i]);
    }
    printf("\nEnter the size of file :\n");
    for (i = 1; i <= nf; i++)
    {
        printf("file %d ", i);
        scanf("%d", &f[i]);
    }
    for (j = 1; j <= nf; j++)
    {
        if (bf[j] == 0) // if bf[j] isn't allocated
        {
            temp = b[j] - f[j];
            if (temp >= 0)
```

if (highest < temp)

S

    frag[i] = p;

    highest = temp;

Q

2

3

frag[i] = highest;

    bf[ff[i]] = 1;

    highest = 0;

3

Printf ("\n file no \t file size \t Block no \t Block - size

\t fragment");

for (p = 1; i < n; j++)

    printf ("\n %d \t %d \t %d \t %d \t %d", fr[i], ff[i],

    b[ff[i]], frag[i]);

getch();

3

Output

Enter no. of blocks : 3

Enter size of each blocks : 5 2 7

Enter no. of process : 1 4

Process no	Process size	Block no.
1	1	3
2	4	1

23/8/23  
N

Enter the number of blocks:3  
Enter the number of files:2

Enter the size of the blocks:  
Block 1:5  
Block 2:2  
Block 3:7  
Enter the size of the files:  
File 1:1  
File 2:4

File_no:	File_size:	Block_no:	Block_size:	Fragment
1	1	1	5	4
2	4	3	7	3

```
PS D:\VS Code\OS> cd "d:\VS Code\OS\" ; if ($?) { gcc fift.c -o fift } ; if ($?) { .\fift }
```

Enter the number of blocks:3  
Enter the number of files:2

Enter the size of the blocks:

Block 1:5  
Block 2:2  
Block 3:7

Enter the size of the files:  
File 1:1  
File 2:4

File No	File Size	Block No	Block Size	Fragment
1	1	2	2	1
2	4	1	5	1

\$ D:\VS Code\OS

cd Algo\OS

1 gcc fit.c

2 ./fit

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286