**ai**.

Main:

Offset of x = -4

Offset of y = -8

Offset of d = -16

Offset of c = -17

g:

Offset of result: -4

Offset of parameter x: +8

f:

Offset of w: -4

Offset of result: -16

Offset of parameter x: +16

Offset of parameter y: +12

Offset of parameter z: +8

**aii**.

Extra space will be needed for padding in main because of the single char, with 7 bytes of padding. Padding must be done at the end of function g’s allocation, having 4 bytes of padding. 4 more bytes of padding must be done to allocate f because of the double result. Thus, there is a total of 15 bytes of padding.

**aiii.**

Main is size 32. g is size 24. f is size 40.

**b.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stack Frame** | **Address** | **Size** | **Contents** | **Purpose** |
| main | 252 | 4 | Pointer somewhere in the OS | Pointer to the next instruction |
| main | 248 | 4 | 0 | Pointer to the previous BR |
| main | 244 | 4 | 1 | Stores local variable x in stack frame |
| main | 240 | 4 | 2 | Stores local variable y in stack frame |
| main | 232 | 8 | Some initialized double variable | Stores local variable d in stack frame. |
| main | 231 | 1 | Some initialized character variable | Local variable c |
| main | 224 | 7 | Garbage | Padding |
| f | 220 | 4 | result | Return address value of f |
| f | 216 | 4 | 1 | Stores parameter x in stack frame |
| f | 212 | 4 | 2 | Stores parameter y in stack frame |
| f | 208 | 4 | -1 | Stores parameter z in stack frame |
| f | 204 | 4 | Line 11 | Instruction to go back to main |
| f | 200 | 4 | 248 | Address of previous BR |
| f | 196 | 4 | 8 | Local variable w |
| f | 192 | 4 | Garbage | Padding |
| f | 184 | 8 | 4.0 | Local variable result |
| g | 180 | 4 | result | Return address value of g |
| g | 176 | 4 | 2 | Stores parameter y in stack frame |
| g | 172 | 4 | Line 23 | Instruction to go back to function f |
| g | 168 | 4 | 200 | Address to previous BR |
| g | 164 | 4 | 2 | Local variable result |
| g | 160 | 4 | Garbage | Padding |