Answer: 3, 2, 15
a[0] = 5, a[1] = 1, a[2] = 15, a[3] = 20, a[4] = 25. int i, j, m; The variable i, j, m are declared as an integer type. i = ++a[1]; becomes i = ++1; Hence i = 2 and a[1] = 2. j = a[1]++; becomes j = 2++; Hence j = 2 and a[1] = 3. m = a[i++]; becomes m = a[2]. Hence m = 15 and i is incremented by 1(i++ means 2++ so i=3). printf("%d, %d, %d", i, j, m); It prints the value of the variables i, j, m.

Answer: 9
x[i] is equivalent to *(x + i),
so (buf + 1)[5] is *(buf + 1 + 5), i.e. buf[6].

Answer: 0 garbage value garbage value
When an array is partially initialized at the time of declaration then the remaining elements of the array is initialized to 0 by default.

Answer: 65480, 65496
The array a[3][4] is declared as an integer array having the 3 rows and 4 columns dimensions. Next, the base address(also the address of the first element) of array is 65472. For a 2D array like a reference to array has type "pointer to array of 4 ints". Therefore, a+1 is pointing to the memory location of first element of the second row in array a. Hence 65472 + (4 ints * 2 bytes) = 65480. Then, &a has type "pointer to array of 3 arrays of 4 ints", totally 12 ints. Therefore, &a+1 denotes "12 ints * 2 bytes * 1 = 24 bytes". Hence, beginning address 65472 + 24 = 65496. So, &a+1 = 65496.

Answer: 10
The sizeof function return the given variable. Example: float a=10; sizeof(a) is 4 bytes. The variable arr is declared as an floating point array and it is initialized with the values. The variable arr has 4 elements. The size of the float variable is 4 bytes. Hence 4 elements x 4 bytes = 16 bytes
sizeof(arr[0]) is 4 bytes. Hence 16/4 is 4 bytes. Hence the output of the program is '4'.

Answer: SSALC C
When i=strlen(str), then it print nothing.
As **i** is with the length of string and hence in array rev the contents are copied as "\0 **SSALC C** ". But while displaying those contents with puts function it stops because of the first character '\0' and hence no output.

Answer: 2022
Explanation: p[3] - p[1] = ASCII value of 'E' - ASCII value of 'A' = 4 . So the expression p + p[3] - p[1] becomes p + 4. So, it will print the string starting from p+4

Answer: 90675
Explanation: address of second character is stored in p and then its value is changed to 0 and hence 8 is replaced by 0.

Answer: String is not empty
printf does not print anything so it will return 0 and hence else part will execute.

Answer: C carefully
Each letter in string[] array is stored in separate address. The starting address in string[] array is stored a pointer variable ptr which is then incremented by 6. Thus "learn " is neglected and "C carefully " is displayed.

Answer: 12
Here, iptr holds the address of first element in an array arr i.e, address of 10. In printf statement we increment the address by (2 * sizeof(int)). Thus, iptr now points to the address of 12. Asterisks (*) in printf statement prints the value inside the address i.e, 12.

Answer: str2 is lesser than str1
The C library function int strncmp(const char *str1, const char *str2, size_t n) is used to compare at most the first n bytes of str1 and str2.